



**Universidad
Católica de
Santa María**

**Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería Electrónica**

**Diseño e implementación de un prototipo de ventilador mecánico operable
manual y remotamente para la atención de pacientes con problemas
respiratorios**

Tesis presentada por:

Teran Vargas, Miguel Angel

ORCID: 0009-0009-6111-7134

Alvarez Gomez, David Joel

ORCID: 0009-0001-2197-1039

para optar el Título Profesional de Ingeniero Electrónico
con Especialidad en Automatización y Control

Asesor (a):

Mg. Delgado Barra, Lucy Angela

ORCID: 0000-0002-7422-4547

Arequipa - Perú

2026

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA ELECTRONICA
CON ESPECIALIDAD EN AUTOMATIZACIÓN Y CONTROL
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 03 de Diciembre del 2025

Dictamen: 013195-C-EPIE-2025

Visto el borrador del expediente 013195, presentado por:

2015701261 - TERAN VARGAS MIGUEL ANGEL

2015247901 - ALVAREZ GOMEZ DAVID JOEL

Titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE VENTILADOR MECÁNICO OPERABLE
MANUAL Y REMOTAMENTE PARA LA ATENCIÓN DE PACIENTES CON PROBLEMAS
RESPIRATORIOS**

Nuestro dictamen es:

APROBADO

Titulo Profesional/Titulo de Segunda Especialidad/Grado Académico a optar:

**INGENIERO ELECTRONICO CON ESPECIALIDAD EN
AUTOMATIZACIÓN Y CONTROL**

**29410027 - SULLA TORRES RAUL RICARDO
DICTAMINADOR**



**29295487 - URRUTIA ESPINOZA MARIO WILLIAM
DICTAMINADOR**



**29720651 - ZEGARRA GAGO HENRY CHRISTIAN
DICTAMINADO**



Diseño e implementación de un prototipo de ventilador mecánico operable manual y remotamente para la atención de pacientes con problemas respiratorios

INFORME DE ORIGINALIDAD

10%	10%	0%	0%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJO DEL ESTUDIANTE

FUENTES PRIMARIAS

1	dspace.espol.edu.ec Fuente de Internet	6%
2	dspace.esPOCH.edu.ec Fuente de Internet	3%
3	repositorio.ug.edu.ec Fuente de Internet	1%
4	desarrollodesoftware.com.co Fuente de Internet	1%

Excluir citas	Apagado	Excluir coincidencias	< 1%
Excluir bibliografía	Apagado		

DEDICATORIA

A mi madre, que siempre demostró su apoyo, cariño y confianza en mí, impulsándome a cumplir con todos los objetivos que me iba trazando durante todo el camino y siempre guiándome para ser una persona de bien.

A mi padre, que a la distancia siempre demostró su apoyo incondicional para poder cumplir todos mis objetivos propuestos.

A mis abuelitos, que siempre creyeron en mí y ahora desde el cielo me guían por el camino del bien.

A mis 2 pequeños de cuatro patas, Negrita y Bobby, que han sido mi compañía en las noches de estudio y trabajo.

Teran Vargas, Miguel Angel

A Dios, por guiar cada uno de mis pasos, tanto en los momentos buenos como en los difíciles, y por brindarme la fortaleza necesaria para alcanzar mis objetivos.

A mi familia, por su apoyo incondicional y por cada sacrificio realizado en las distintas etapas de mi vida. Gracias por ser mi mayor motivación para seguir creciendo y no rendirme ante los desafíos. Este logro es para ustedes.

A mis compañeros de trabajo, por estar siempre pendientes de mis avances e impulsarme a continuar desarrollándome tanto en el ámbito laboral como académico.

Álvarez Gómez, David Joel

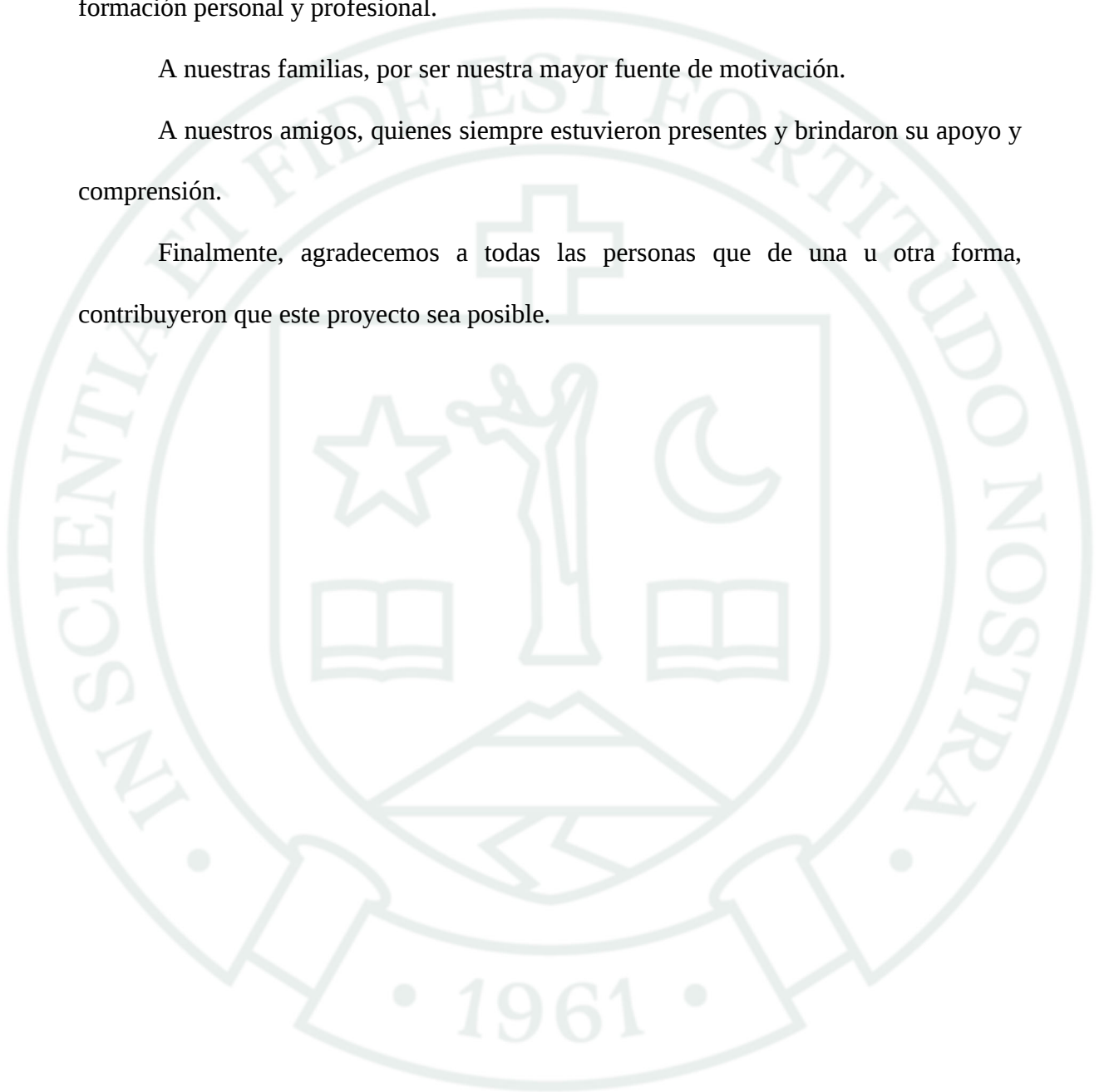
AGRADECIMIENTOS

A nuestros profesores y compañeros de la Universidad Católica de Santa María, gracias por compartir conocimientos, experiencias y momentos que enriquecieron mi formación personal y profesional.

A nuestras familias, por ser nuestra mayor fuente de motivación.

A nuestros amigos, quienes siempre estuvieron presentes y brindaron su apoyo y comprensión.

Finalmente, agradecemos a todas las personas que de una u otra forma, contribuyeron que este proyecto sea posible.



RESUMEN

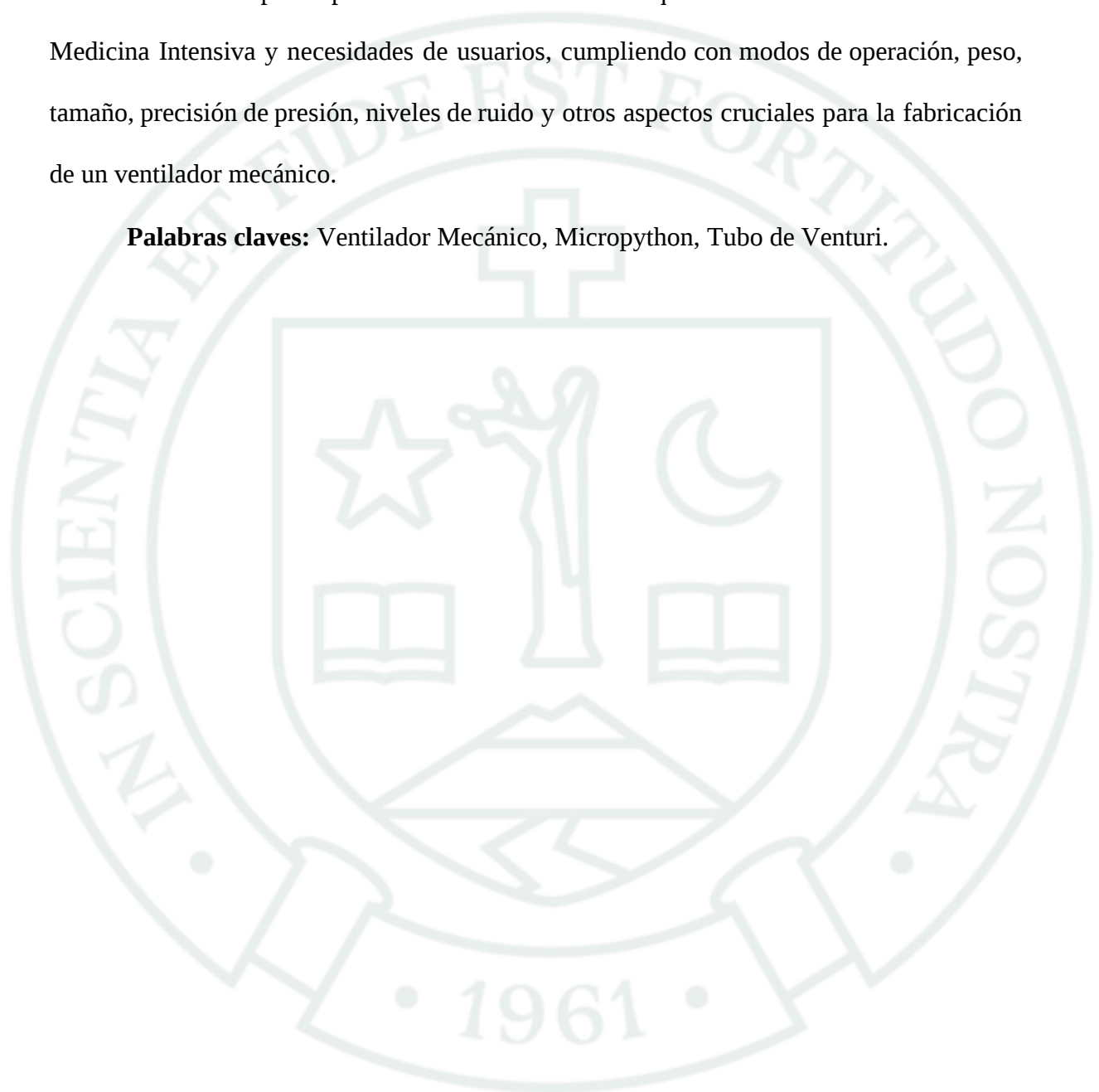
En este trabajo de investigación, se pretende diseñar e implementar un prototipo de ventilador mecánico para la atención de pacientes con problemas respiratorios, que pueda operar tanto de forma manual como remota, adaptándose a las necesidades de pacientes adultos e infantes. Tras un estudio de necesidades realizado durante la pandemia de COVID-19, en colaboración con trabajadores de la salud y expertos en ingeniería de la UCSM, se identificaron criterios clave como un presupuesto asequible, capacidad para medir la presión de aire, tamaño compacto y niveles de ruido bajos. Basándose en estas consideraciones, se diseñó un ventilador mecánico en 3D, inspirado en la bolsa de resucitación Ambu, con dimensiones de 40x13x9 cm y un peso inferior a 8 kg.

El sistema fue desarrollado utilizando dos microcontroladores Raspberry Pi Pico, los cuales fueron programados en Micropython mediante el entorno de desarrollo Thonny IDE. Estos microcontroladores se encargan de gestionar los distintos modos de operación (infante-adulto) y las modalidades de control (local-remoto) se da mediante los módulos Bluetooth CH-06 y USB Serial RS232, donde se utilizó el Driver TB6600 para el control del motor paso a paso, y se empleó el sensor MPX5010DP para la captura de la presión. Se creó una interfaz gráfica de usuario en Visual Basic para el control de modos de operación, y el registro de la hora, la presión y el flujo de aire. Asimismo, se desarrolló una aplicación para dispositivos Android mediante MIT App Inventor, este último controla al ventilador inalámbricamente a través de Bluetooth.

Las pruebas se llevaron a cabo utilizando un pulmón artificial, luego el prototipo se reguló a los estándares de respiración para infantes (6-12 años a 22.5 rpm) y adultos (65 años o más a 14 rpm). Se empleó el principio del tubo de Venturi y el sensor con diferentes configuraciones de operación del motor, utilizando el método de clasificación de árbol de decisión se seleccionó dos configuraciones que se aproximen a los estándares de

respiración de infantes y adultos, manteniendo bajos niveles de ruido. Durante las pruebas, se lograron valores de presión de 15.25 cmH₂O a 21.66 rpm con una desviación de 0.46 cmH₂O en modo infantil, y 19.96 cmH₂O a 14.77 rpm con una desviación de 0.46 cmH₂O en modo adulto. El prototipo fue validado conforme a requisitos de la Sociedad Chilena de Medicina Intensiva y necesidades de usuarios, cumpliendo con modos de operación, peso, tamaño, precisión de presión, niveles de ruido y otros aspectos cruciales para la fabricación de un ventilador mecánico.

Palabras claves: Ventilador Mecánico, Micropython, Tubo de Venturi.



ABSTRACT

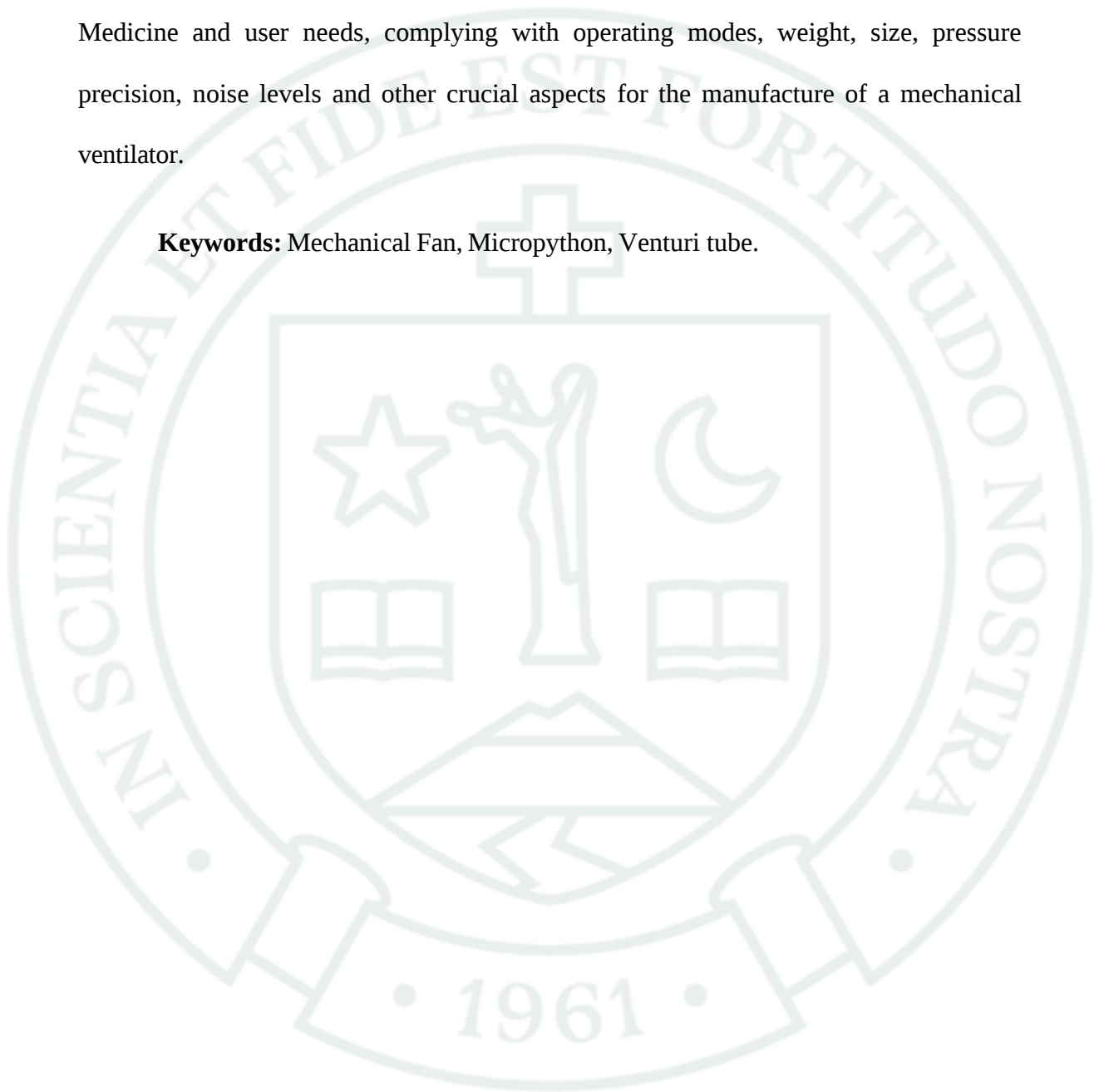
In this research work, the aim is to design and implement a mechanical ventilator prototype for the care of patients with respiratory problems, which can operate both manually and remotely, adapting to the needs of adult and infant patients. Following a needs study conducted during the COVID-19 pandemic, in collaboration with UCSM healthcare workers and engineering experts, key criteria were identified such as an affordable budget, ability to measure air pressure, compact size, and levels low noise. Based on these considerations, a 3D mechanical ventilator was designed, inspired by the Ambu resuscitation bag, with dimensions of 40x13x9 cm and a weight of less than 8 kg.

The system was developed using two Raspberry Pi Pico microcontrollers, which were programmed in Micropython using the Thonny IDE development environment. These microcontrollers are responsible for managing the different operating modes (infant-adult) and the control modes (local-remote) are given through the Bluetooth CH-06 and USB Serial RS232 modules, where the TB6600 Driver was used to control the stepper motor, and the MPX5010DP sensor was used to capture the pressure. A graphical user interface was created in Visual Basic to control operating modes, and record time, pressure and air flow. Likewise, an application for Android devices was developed using MIT App Inventor, the latter controls the fan wirelessly via Bluetooth.

The tests were carried out using an artificial lung, then the prototype was calibrated to breathing standards for infants (6-12 years at 22.5 rpm) and adults (65 years or older at 14 rpm). The Venturi tube principle and MPX5010DP sensor were used to measure air pressure and flow. Extensive tests were carried out with different motor operation configurations, using the decision tree classification method, two configurations were selected that approximate the breathing standards of infants and

adults, while maintaining low noise levels. During the tests, pressure values of 15.25 cmH₂O at 21.66 rpm with a deviation of 0.46 cmH₂O in child mode, and 19.96 cmH₂O at 14.77 rpm with a deviation of 0.46 cmH₂O in adult mode were achieved. The prototype was validated in accordance with the requirements of the Chilean Society of Intensive Medicine and user needs, complying with operating modes, weight, size, pressure precision, noise levels and other crucial aspects for the manufacture of a mechanical ventilator.

Keywords: Mechanical Fan, Micropython, Venturi tube.



ÍNDICE

DEDICATORIA

AGRADECIMIENTOS

RESUMEN

ABSTRACT

INTRODUCCIÓN 1

CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA 3

1.1 Objetivos del Estudio 4

1.1.1 *Objetivo General* 4

1.1.2 *Objetivos Específicos* 4

1.2 Justificación del Estudio 5

1.3 Alcances y Limitaciones 6

1.3.1 *Alcances* 6

1.3.2 *Limitaciones* 7

1.4 Metodología 7

CAPITULO II MARCO TEÓRICO 9

2.1 Antecedentes 9

2.1.1 *Antecedentes Internacionales* 9

2.1.2 *Antecedentes Nacionales* 10

2.2 Bases Teóricas 11

2.2.1 *El Aparato Respiratorio* 11

2.2.2 *Fisiología Respiratoria y Transporte de Oxígeno* 11

2.2.3 *Modalidades de Ventilación Mecánica Pulmonar* 13

2.2.4 *Ventilación Mecánica* 15

2.2.5 *Medición de Flujo y Presión de Aire (en tubos)* 16

2.2.6 *Teorema de Bernoulli* 16

2.2.7 *Sensores, reguladores de aire y oxígeno Sensor de presión de aire* 18

2.2.8 *Microcontroladores* 20

2.2.9	Lenguajes de Programación.....	25
2.2.10	Entorno de Desarrollo (IDE)	28
2.3	Marco Conceptual Biológico	30
2.3.1	Asimetrías de Ciclos de Inhalación y Ciclos de Respiración	30
2.3.2	Frecuencia Respiratoria	31
2.3.3	Humidificador	32
2.3.4	Presión Pico.....	33
2.3.5	Presión Residual.....	34
2.3.6	Volumen Tidal.....	35
2.4	Marco Conceptual Tecnológico.....	35
2.4.1	Fritzing.....	35
2.4.2	APP.....	36
2.4.3	SolidWorks	37
CAPITULO III METODOLOGIA Y DIMENSIONAMIENTO DEL PROTOTIPO		38
3.1	Metodología de Diseño	38
3.1.1	Requerimientos del Usuario.....	38
3.1.2	Punto de Vista de Fabricación.....	40
3.1.3	Estudio de la Competencia.....	43
3.1.4	Estructura Funcional	43
3.1.5	Conceptualización de las Posibles Soluciones	46
3.1.6	Criterios para la Selección de Soluciones	49
3.1.6.1	Diseño compacto	50
3.1.6.2	Funcionamiento sencillo	51
3.1.6.3	Alta durabilidad	51
3.1.6.4	Costo de operación	52
3.1.6.5	Costo de manufactura	53
3.1.6.6	Seguridad.....	53
3.1.7	Conclusiones y Selección de Alternativa	54

3.1.8	Criterios de la Estructura Impulsor de Aire	55
3.2	Dimensionamiento y Diseño del Tornillo sin fin.....	56
3.2.1	Dimensionamiento del Tornillo sin fin	56
3.2.1.1	Esfuerzo a la tensión.....	58
3.2.1.2	Esfuerzo cortante.....	58
3.2.1.3	Esfuerzo a la torsión	59
3.2.1.4	Esfuerzo flexionante en la raíz de la rosca.....	60
3.3	Dimensionamiento del Motor	61
3.4	Dimensionamiento del Método de Tubo de Venturi	62
3.4.1	Dimensionamiento.....	62
3.5	Instrumentación del Ventilador Mecánico.....	65
3.6	Sistema de Control del Ventilador	67
3.7	Adquisición de Dispositivos Biomédicos y Electrónicos.....	68
3.7.1	Bolsa de Ambu	68
3.7.2	Guantes Látex.....	70
3.7.3	Regulador de Oxígeno	70
3.7.4	Sensor de Presión Diferencial.....	71
3.7.5	Motor Paso a Paso.....	71
3.7.6	Tornillo Guía sin fin.....	72
3.7.7	Sensor de Final de Carrera.....	73
3.7.8	Driver de Control.....	74
3.7.9	Raspberry Pi Pico	74
3.7.10	Display Oled	75
3.7.11	Módulo Bluetooth.....	76
3.7.12	Módulo Serial USB	76
3.7.13	Tubo Flexible de 1/8 y 1 ½	77
3.7.14	Fuente de Alimentación Conmutada	78
3.8	Cumplimiento de las normas.....	78

3.8.1 Norma ISO 10651-4	78
3.8.2 Norma ISO 80601-2-12:2020	81
3.8.3 NTP IEC 60601-1	83
3.9 Criterios de selección de componentes electrónicos	84
CAPITULO IV DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO	93
4.1 Diseño e impresión en 3D.....	93
4.1.1 Tubo de Venturi	94
4.1.2 Impulsor de Aire.....	96
4.1.3 Control de mando	102
4.2 Implementación Estructural del Prototipo	104
4.2.1 Extractor de Aire, Bolsa Ambu	105
4.2.2 Control de Mando.....	106
4.2.3 Tubo de Venturi	107
4.3 Implementación Eléctrica, Electrónica del Prototipo	108
4.3.1 Local o Manual	111
4.3.2 Modo Remoto.....	113
4.4 Programación de los Microcontroladores con Micropython.....	114
4.4.1 Control Manual o Local.....	116
4.4.2 Control Remoto.....	122
4.5 Desarrollo y Programación de GUI en Visual Basic	128
4.6 Desarrollo de la APP	133
CAPITULO V PRUEBAS Y RESULTADOS	136
5.1 Calibración del prototipo.....	137
5.1.1 Testeo y adquisición de data.....	138
5.1.2 Consideraciones Médicas.....	141
5.1.3 Calibración modo infante	141
5.1.4 Calibración modo adulto	143
5.2 Prueba de funcionamiento del prototipo	145

5.2.1	<i>Prueba de los controles de operación</i>	147
5.2.2	<i>Prueba de la presión de aire y volumen</i>	150
5.3	Resultado de presión y rpm.....	153
5.3.1	<i>Modo Infante</i>	153
5.3.2	<i>Modo Adulto</i>	156
5.4	Fiabilidad del prototipo	158
5.5	Análisis Económico y Presupuesto del Prototipo	161
5.5.1	<i>Análisis Económico.....</i>	161
5.5.2	<i>Presupuesto del Prototipo.....</i>	161
CONCLUSIONES.....		166
RECOMENDACIONES.....		170
BIBLIOGRAFÍA.....		172
ANEXOS.....		176

ÍNDICE DE FIGURAS

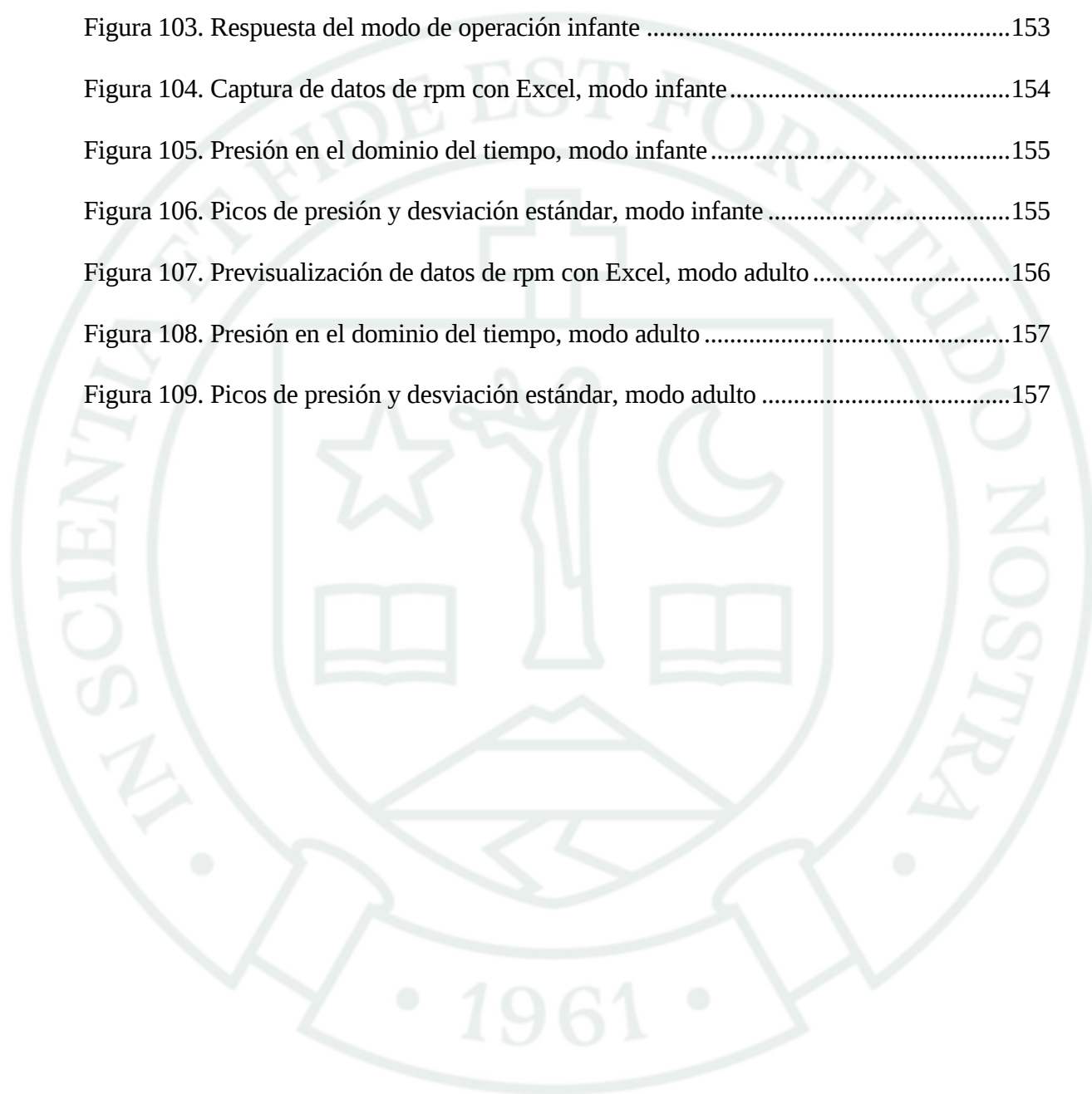
Figura 1. Cifras de afectos por Covid-19	3
Figura 2. Esquema del prototipo a diseñar	7
Figura 3. Efecto Bohr	12
Figura 4. Curva de disociación de la hemoglobina	13
Figura 5. Ventilación controlada por volumen, presión y mixta	14
Figura 6. Curvas de presión-tiempo	14
Figura 7. SIMV por volumen, curva de presión-tiempo	15
Figura 8. Sensor de presión de aire.....	18
Figura 9. Sensor de flujo de aire.....	19
Figura 10. Regulador de oxígeno	20
Figura 11. Arquitectura del microcontrolador	21
Figura 12. Diagrama de los pines.....	22
Figura 13. Chip del sistema RP2040	24
Figura 14. Ubicación de los pines	25
Figura 15. Logotipo Visual Estudio	28
Figura 16. Interfaz en App Inventor	29
Figura 17. Thonny	30
Figura 18. Mapa Conceptual de la metodología de diseño	38
Figura 19. Función global del ventilador mecánico	44
Figura 20. Análisis de funcionamiento del ventilador mecánico	45
Figura 21. Ventilador con turbina centrífuga	46
Figura 22. Ventilador con leva	47
Figura 23. Ventilador con tornillo sin fin.....	47
Figura 24. Diseño de la bolsa de Ambu en SolidWorks.....	55
Figura 25. Ventilador artificial con tornillo sin fin.	56

Figura 26. Tornillo sin fin incrustado a un Servomotor	56
Figura 27. Diagrama de Cuerpo Libre (DCL) del tornillo del tornillo sin fin.....	57
Figura 28. Tubo de Venturi	62
Figura 29. Tubo de Bernoulli	63
Figura 30. Esquema de dispositivos electrónicos del Ventilador mecánico.....	66
Figura 31. Diagrama de flujo de control y operación de ventilador.....	67
Figura 32. Bolsa de Ambu.....	70
Figura 33. Guante quirúrgico talla M.....	70
Figura 34. Regulador de oxígeno YH. YX11A.....	71
Figura 35. Sensor MPX5010 DP.....	71
Figura 36. Motor Nema 17 de paso a paso	72
Figura 37. Tornillo sin fin (CNC)	73
Figura 38. Sensor final de carrera mecánico	74
Figura 39. Driver de control	74
Figura 40. Raspberry Pi Pico.....	75
Figura 41. Display Oled.....	75
Figura 42. Módulo Bluetooth	76
Figura 43. Modulo serial USB.....	77
Figura 44. Tubo flexible de 1/8 y 1 ½.....	77
Figura 45. Fuente de alimentación conmutada	78
Figura 46. Plano y vistas del tubo de Venturi.	95
Figura 47. Impresión del tubo de Venturi	96
Figura 48. Plano de las piezas del impulsor de aire.....	97

Figura 49. Estructura de la presión de la bolsa Ambu.....	98
Figura 50. Impresión de la pieza V3-1	102
Figura 51. Plano y vistas del control de mando.....	103
Figura 52. Piezas del control de mando	104
Figura 53. Ensamblado3D del prototipo en SolidWorks.....	104
Figura 54. Extractor de aire diseñado en SolidWorks	105
Figura 55. Extractor de aire montado	106
Figura 56. Control de mando.....	107
Figura 57. Empalme del tubo de Venturi	107
Figura 58. Esquemático de los componentes electrónicos	109
Figura 59. Esquema electrónico de conexión y alimentación para el prototipo	110
Figura 60. Distribución de pines de Raspberry Pi Pico	111
Figura 61. Conexión para modo manual	112
Figura 62. Conexión con Driver TB6600.....	113
Figura 63. Conexión para el modo de operación remoto	114
Figura 64. Captura de pantalla de la programación en IDE de Thonny	115
Figura 65. Diagrama de flujo de la operación y control de microcontroladores.....	116
Figura 66. Diagrama de flujo, modo manual.....	117
Figura 67. Importación de librerías, Micropython	118
Figura 68. Declaración de variables, Micropython	118
Figura 69. Funciones de interrupción, Micropython.....	119
Figura 70. Bucle While True, Micropython	119
Figura 71. Función modo adulto, Micropython	120
Figura 72. Función modo infante, Micropython	121
Figura 73. Diagrama de flujo, modo remoto parte 1.....	123
Figura 74. Diagrama de flujo, modo remoto parte 2.....	124

Figura 75. Importación de librerías y funciones, Micropython	125
Figura 76. Declaración de parámetros y puertos, Micropython.....	125
Figura 77. Función ADS1X15, Micropython.....	126
Figura 78. Función de lectura I2C, Micropython.....	126
Figura 79. Operación de presión, Micropython	127
Figura 80. Lectura de puertos parte 1, Micropython.....	128
Figura 81. Lectura de puertos parte 2, Micropython.....	128
Figura 82. Gráfico de usuario del prototipo	129
Figura 83. Función conectar, Visual Basic.....	130
Figura 84. Función actualizar, Visual Basic.....	130
Figura 85. Escalar los datos, Visual Basic.....	131
Figura 86. Función botones, Visual Basic.....	131
Figura 87. Función de almacenamiento, Visual Basic.....	132
Figura 88. Interfaz gráfico de usuario para teléfono móvil.....	133
Figura 89. Conexión mediante bluetooth, APP Inventor	134
Figura 90. Lectura y escritura de datos seriales, APP Inventor	135
Figura 91. Fotografías del prototipo	136
Figura 92. Configuración del driver y del código	137
Figura 93. Sonómetro y generación de gráficos con Excel.....	138
Figura 94. Método de árbol de clasificación del modo infante.....	142
Figura 95. Distribución de ruido y rpm modo infante.....	143
Figura 96. Método de árbol de clasificación del modo adulto.....	144
Figura 97. Distribución de ruido y rpm de modo adulto.....	145
Figura 98. Fotografías de las pruebas del prototipo	146

Figura 99. Pulsadores de modo de operación local	148
Figura 100. Pantalla de control inalámbrico.....	149
Figura 101. Control por comunicación serial USB	150
Figura 102. Respuesta del modo de operación adulto.....	151
Figura 103. Respuesta del modo de operación infante	153
Figura 104. Captura de datos de rpm con Excel, modo infante	154
Figura 105. Presión en el dominio del tiempo, modo infante	155
Figura 106. Picos de presión y desviación estándar, modo infante	155
Figura 107. Previsualización de datos de rpm con Excel, modo adulto	156
Figura 108. Presión en el dominio del tiempo, modo adulto	157
Figura 109. Picos de presión y desviación estándar, modo adulto	157



ÍNDICE DE TABLAS

Tabla 1. Cuadro de las características más importantes en los ventiladores mecánicos desde el punto de vista del usuario	39
Tabla 2. Cuadro de relación entre requerimientos del usuario y requisitos funcionales	41
Tabla 3. Medios para satisfacer los requerimientos planteados para el diseño.....	42
Tabla 4. Especificaciones técnicas generales de ventiladores respiratorios, Perú	43
Tabla 5. Matriz morfológica.....	49
Tabla 6. Importancia en escala de 0 a 5 de cada solución.....	50
Tabla 7. Ponderación de preferencia de cada solución con relación al diseño compacto	51
Tabla 8. Ponderación de preferencia de cada solución con relación al funcionamiento sencillo.....	51
Tabla 9. Ponderación de preferencia de cada solución con relación a alta durabilidad	52
Tabla 10. Ponderación de preferencia de cada solución con relación al costo de operación	52
Tabla 11. Ponderación de preferencia de cada solución con relación costo de manufactura	53
Tabla 12. Ponderación de preferencia de cada solución con relación a la seguridad del paciente	53
Tabla 13. Conclusiones y selección de la mejor alternativa.....	54
Tabla 14. Características del tornillo guía sin fin.....	57
Tabla 15. Cumplimiento de la norma ISO 10651-4 en el prototipo.....	80

Tabla 16. Cumplimiento de la norma ISO 80601-2-12:2020 en el prototipo	83
Tabla 17. Cumplimiento de la norma NTP IEC 60601-1 en el prototipo	84
Tabla 18. Relación entre criterios de selección y componentes.....	86
Tabla 19. Características de las posibles alternativas de microcontrolador	87
Tabla 20. Relación entre criterios de selección y las alternativas de microcontrolador	87
Tabla 21. Características de las posibles alternativas de sensor de presión.....	88
Tabla 22. Relación entre criterios de selección y las alternativas de sensor de presión	88
Tabla 23. Características de las posibles alternativas de conversor A/D	89
Tabla 24. Relación entre criterios de selección y las alternativas de conversor A/D	89
Tabla 25. Características de las posibles alternativas de motor	90
Tabla 26. Relación entre criterios de selección y las alternativas de motor.....	90
Tabla 25. Características de las posibles alternativas de motor	90
Tabla 26. Relación entre criterios de selección y las alternativas de motor.....	90
Tabla 27. Características de las posibles alternativas de driver	91
Tabla 28. Relación entre criterios de selección y las alternativas de driver.....	91
Tabla 29. Características de las posibles alternativas de pantalla	92
Tabla 30. Relación entre criterios de selección y las alternativas de pantalla	92
Tabla 31. Propuesta de materiales del prototipo ventilador mecánico.....	98
Tabla 32. Características del extractor de aire	106
Tabla 33. Datos generados a partir de los indicadores.....	140
Tabla 34. Respiración por minuto según la edad	141
Tabla 35. Configuración hardware y software de modos de operación.....	146
Tabla 36. Estado de operación	147
Tabla 37. Fiabilidad del prototipo de ventilador mecánico.....	158
Tabla 38. Presupuesto de materiales del prototipo ventilador mecánico	163
Tabla 39. Presupuesto de mano de obra del prototipo ventilador mecánico.....	165

ÍNDICE DE ECUACIONES

Ecuación 1.....	16
Ecuación 2.....	17
Ecuación 3.....	18
Ecuación 4.....	57
Ecuación 5.....	57
Ecuación 6.....	57
Ecuación 7.....	58
Ecuación 8.....	59
Ecuación 9.....	59
Ecuación 10.....	60
Ecuación 11.....	61
Ecuación 12.....	61
Ecuación 13.....	63
Ecuación 14.....	64
Ecuación 15.....	64
Ecuación 16.....	152
Ecuación 17.....	152

INDICE DE ANEXOS

Anexo A: Ventiladores Respiradores, Perú.....	176
Anexo B: Características y hoja de datos de los Componentes	180
Anexo B1: Bolsa Ambu	180
Anexo B2: Sensor MPX5010DP.....	182
Anexo B3: Motor nema 17.....	185
Anexo B4: Tornillo Sin fin CNC	187
Anexo C: Diseño de componentes y piezas.	189
Anexo C1: Bolsa de Ambu.....	189
Anexo C2: Motor Mena 17 y tornillo de guía sin fin.....	190
Anexo C3: Tubo de Venturi.....	191
Anexo C4: Estructura de compresión explosionada.....	192
Anexo D: Estructura del Prototipo	193
Anexo D1: Vistas en el plano ISO_E.....	193
Anexo D2: Vista explosiva de la estructura	194
Anexo E: Raspberry Pi Pico R3.....	195
Anexo F: Código	199
Anexo F1: Micropython R1	199
Anexo F2: Micropython R2	202
Anexo F3: Visual Basic (GUI).....	206
Anexo F4: Aplicación en APP Inventor.....	213
Anexo G: Prueba	216
Anexo G1: Promedio Fisiológico de respiración	216
Anexo G2: Tabla experimental	217
Anexo G3: Captura de pantalla de la tabla experimental.....	219
Anexo H: Formulario de evaluación y encuesta	224

Anexo I: Circuito PCB y plano esquemático	228
Anexo J: Norma y estándar	230
Anexo K: Google Colab.....	232
Anexo L: Constancia de evaluación de fiabilidad del prototipo.....	234
Anexo M: Especificaciones técnicas y físicas del prototipo.....	232
Anexo N: Condiciones de operación y parámetros.....	234



INTRODUCCIÓN

La demanda de equipos de soporte vital en hospitales y centros de salud, aumento a partir de la pandemia del COVID-19, evidenciando una clara falta de preparación ante este tipo de situaciones, como ejemplo tenemos la carencia de ventiladores mecánicos y la falta de capacitación para el manejo, mantenimiento del mismo y la alta inversión monetaria que es necesaria para adquirir este tipo de equipos , en Arequipa a inicios de la pandemia solo se contaban con 17 ventiladores mecánicos cuando la demanda exigía una cantidad mucho mayor originando falencias (Gobierno Regional de Arequipa, 2021).

Los pacientes que requieren estos dispositivos médicos se encuentran en la necesidad de adquirirlos, ya que los hospitales estatales carecen de suficientes unidades, lo que se convierte en una barrera de acceso para muchos, en especial para niños y adultos mayores.

Por tanto, este proyecto asume el desafío de diseñar e implementar un ventilador mecánico con dos modos de operación: uno para niños y otro para adultos. Además, se plantea el desarrollo de dos modos de control: uno remoto y otro manual.

Para lograr nuestros objetivos, se busca nuevas alternativas de solución en función mecánica y programación, es por ello que la programación se realizó en Micropython usando los módulos Raspberry Pi Pico, siendo este el controlador de la automatización del ventilador mecánico en función a la presión y caudal de aire que se genera.

En el Capítulo 1 se describe el planteamiento del problema, los objetivos planteados, la justificación de la investigación, los alcances y limitaciones, posteriormente la metodología de secuencia.

En el Capítulo 2 se detalla la recopilación de antecedentes y el marco teórico de la fisiología de respiración humana, así como del ventilador mecánico.

En el Capítulo 3 se especifica el metodología y dimensionamiento para el desarrollo del ventilador, donde se realizó encuestas a expertos de la materia para considerar las características que debe de tener el ventilador. Además, la selección y adquisición de los componentes.

En el Capítulo 4, se desarrolla el diseño y la implementación de la estructura, se implementa la parte eléctrica, también la programación de los microcontroladores en Micropython, y los interfaces de usuario GUI, en App Inventor y Visual BASIC.

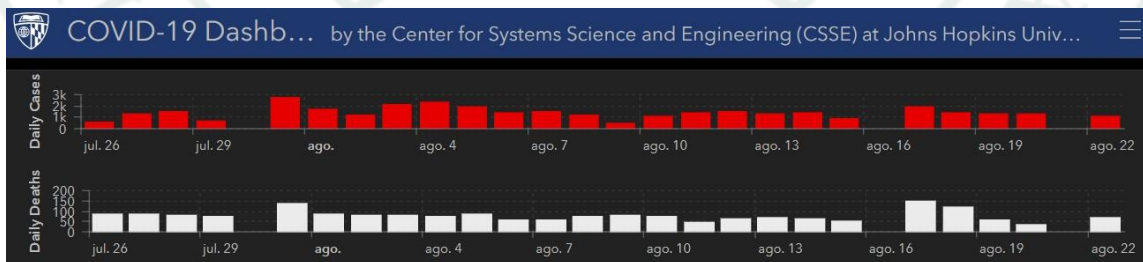
En el Capítulo 5, se llevan a cabo pruebas exhaustivas del funcionamiento del ventilador mecánico en ambos modos de operación y con los tres tipos de control disponibles. Durante estas pruebas, se registra la data de presión y se analizan los picos más altos de presión para estimar la respiración por minuto según la edad del paciente. Además, se evalúa la fiabilidad del prototipo, asegurando que cumpla con los estándares establecidos y los requisitos específicos del usuario.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Teniendo en cuenta la anterior crisis sanitaria debido al COVID-19 donde se evidencio la falta de equipo de soporte vital en las unidades de cuidados intensivos, actualmente seguimos afrontando el desabastecimiento tanto de equipos como de personal, con lo cual podemos deducir que no estamos preparados para otra crisis sanitaria, según la OCDE la recomendación de unidades de cuidados intensivos es de 12 unidades por cada 100,000 personas pero nuestro país afronto la pandemia con una relación de 1 a 100,000 siendo la relación más baja en Latinoamérica, debemos tener en cuenta también que la tasa de recuperación de pacientes en cuidados intensivos es aproximadamente de 1 a 7 días en los cuales necesita equipo de soporte vital y personal que lo atienda, cabe resaltar que hasta la actualidad no se han empleado medidas para aumentar significativamente las capacidades de estas unidades en las cuales se encuentran equipos como ventiladores mecánicos, monitores, bombas de infusión, etc, los cuales son equipos de precios muy elevados.

Figura 1
Cifras de afectos por Covid-19



Nota. JHU CSSE COVID-19 Data.

Como se puede apreciar, los principales problemas relacionados al trabajo planteado fueron el elevado precio de los ventiladores mecánicos y la cantidad de estos, los que en momentos de mayor necesidad fueron escasos no solo a nivel del país, sino en todo el mundo.

Según MINSA, a la fecha contamos con 1483 ventiladores mecánicos operativos para uso COVID y no COVID (MINSA, 2024), de los que a inicios de la pandemia solo se contaba con aproximadamente 100 y en poco tiempo se adquirieron más, pero a precios elevados (Diario la Republica, 2022). En un caso de costos, el Perú a la adquisición de 485 unidades de ventiladores volumétricos mecánicos adulto/pediátrico, los que tuvieron un precio aproximado de S/.71.000.000,00 en total. (Resolución Ministerial N° 260-2022-MINSA), resaltando el gran costo de este tipo de dispositivos, que en general llegan a precios menores como 12.000 USD, pero que sin embargo están fuera del alcance de las personas promedio, que por necesidad algunas llegaron a diseñar modelos poco confiables. La asistencia de participantes con modus operandi muy diverso, así como la diligencia a la que ha crecido el campo, llevaron a un espacio de diseño segmentado determinado por la ausencia de patrones de diseño claros, planificaciones que no respetan con los requerimientos funcionales mínimos o muestran poco o nada Innovación (Mora et al., 2020).

1.1 Objetivos del Estudio

1.1.1 Objetivo General

- Diseñar e implementar un prototipo de ventilador mecánico operable manual y remotamente para atención de pacientes con problemas respiratorios.

1.1.2 Objetivos Específicos

- Comprender el funcionamiento de los ventiladores mecánicos.
- Determinar los requisitos necesarios del ventilador mecánico para definir los parámetros de diseño del mismo.
- Seleccionar los componentes adecuados de acuerdo al diseño establecido.
- Implementar un ventilador que cumpla con las normas ISO 10651-4 en cuanto

a los requisitos técnicos y de seguridad establecidos.

- Evaluar la confiabilidad y desempeño a lo largo del tiempo para garantizar su durabilidad y fiabilidad.
- Establecer una conexión inalámbrica mediante bluetooth entre el prototipo y una interfaz de operación remota diseñada para celulares. Esta interfaz permitirá la monitorización visual de la respiración del paciente y el control del ventilador en los modos para niños y adultos.
- Controlar de forma local los dos modos de operación del prototipo.
- Obtener valores de presión y volumen, e implementar un sistema de almacenamiento de datos que permita interpretar y obtener información valiosa sobre la respiración del paciente.

1.2 Justificación del Estudio

El mayor problema viene dado por la contingencia que se dió a nivel mundial, la cual dejó estragos a nivel mundial y resaltando el déficit de cantidad de este tipo de dispositivos (diseño y evaluación de un ventilador mecánico), y más aún a nivel nacional. El contar con este tipo de sistemas no solo ayudan por la pandemia, ya que al ser un sistema de soporte vital puede ayudar en otro tipo de escenarios. También a futuro podrían acontecer otros problemas que como se ve las enfermedades actuales van transmutando. Como menciona Lawrence Young, profesor de Oncología Molecular en Warwick Medical School, "Habrá futuras pandemias. Es decir, claramente todo ha cambiado y este ha sido un punto de giro en términos de cómo veremos las pandemias en el futuro y cómo vamos a necesitar estar mucho más preparados" (Sanchez, 2021).

Los ventiladores mecánicos en su mayoría se encuentran diseñados para la operación y programación en el mismo ventilador (Husseini et al., 2010), pero poder añadir la operabilidad por un medio inalámbrico es un gran aporte, además el diseñar el

dispositivo ayuda a dimensionarlo en base a las necesidades propias, porque al usar controladores, módulos y sensores, el prototipo se tiene que acondicionar a lo necesario, y así darle robustez, como se busca en dispositivos de esta índole que son de uso médico.

1.3 Alcances y Limitaciones

1.3.1. Alcances

El prototipo comprenderá la implementación de un sistema diseñado en un software de modelado 3D que pueda realizar la función mecánica para ejercer movimiento en la bolsa AMBU por medio del uso de un motor de pasos, todo esto basado en los parámetros que se ingresan, ya que también contará con un sistema de pulsadores y una pequeña pantalla para un control manual pero también con un sistema de escritorio para computadoras, que al contar con un monitor se puede tener otro tipo de visualización del funcionamiento.

El prototipo se caracteriza por ser de bajo costo y utilizar software de código abierto, lo cual permite que el código fuente pueda ser inspeccionado, modificado y mejorado por cualquier persona. Se utilizarán lenguajes de programación como Micropython, Visual Basic y aplicaciones como el MIT App Inventor.

El ventilador mecánico debe ser portátil y no pesar más de 10 kg. También debe tener una fuente de alimentación de 220V y sus dimensiones deben ser menores a 80 cm por lado. Además, debe ser controlado en 3 modos (1 local y 2 remotos) y tener 2 modos de funcionamiento, un volumen tidal de entre 300 ml a 600 ml, frecuencia respiratoria de 10 a 40 respiraciones por minuto y una presión positiva final de la espiración entre 5 y 25 cmH₂O con una precisión de 1 cmH₂O. Es importante destacar que el ventilador funcionará de manera independiente, es decir, no dependerá de un computador una vez que esté en marcha.

1.3.2. Limitaciones

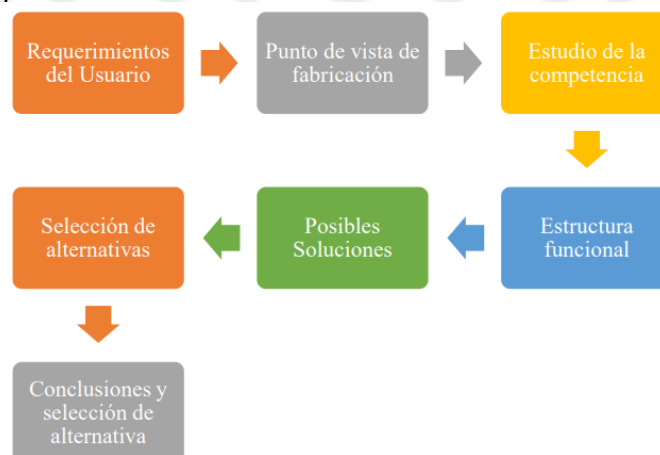
Una de las principales limitaciones del prototipo se encuentra en el tipo de material utilizado en la construcción del dispositivo. Algunos de los materiales involucrados no cuentan con certificación ISO 13485 e IEC 60601-1 en cuanto a bioseguridad para el COVID-19. Por lo tanto, no se pueden realizar pruebas con pacientes y se utilizarán pulmones de maniquí en su lugar. La implementación de un ventilador mecánico puede resultar costosa, por lo que se optó por fabricar un prototipo de costo económico que cumpla con las funciones y parámetros de funcionamiento de un ventilador mecánico certificado. Otro aspecto limitado es el diseño de la interfaz de control por computadora y celular. Si bien cumple con las funciones básicas para los modos de adultos y niños en el control y almacenamiento, se ve limitada en aspectos como el aprendizaje automático o las alarmas.

1.4 Metodología

Se trata de una investigación aplicada que busca encontrar una estrategia que permita diseñar y elaborar un prototipo de ventilador mecánico de costo rentable que permita la visualización de información vital y la operación de este por diferentes modos.

Figura 2

Esquema del prototipo a diseñar



Nota. Elaboración propia.

Primero se realizará una consulta a profesionales de la salud (doctores, enfermeros, técnicos de enfermería, técnicos de mantenimiento de equipos médicos, etc.) y pacientes que usaron este tipo de equipos, con la finalidad de identificar características básicas que debe cumplir el prototipo. En segundo lugar, se realizará una consulta con especialistas relacionados con equipos mecánicos y electrónicos de los cuales esperamos identificar algunas características de funcionamiento, mantenimiento y dimensiones del prototipo. En tercer lugar, se buscará información de ventiladores mecánicos ya existentes operativos en el mercado tanto de diseño independiente o privado con la finalidad de hallar el costo que podría alcanzar nuestro prototipo.

En base a las encuestas realizadas definiremos una estructura funcional a la cual agregaremos las etapas que sean necesarias para que el prototipo pueda cumplir con las conclusiones y objetivos. Así poder crear un listado de posibles soluciones de las cuales se escogerá la mejor alternativa.

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes

2.1.1 Antecedentes Internacionales

Flores (2020), en su trabajo de investigación contempla como objetivos el poder diseñar y fabricar prototipos de ventiladores de forma mecánica que va a servir de como sistema respiratorio en épocas de pandemia producto de la COVID-19, asimismo, el poder realizar ensayos para verificar parámetros electrónicos para el análisis de riesgo del prototipo. Cuyo diseño de estudio está enfocado en la tecnología biomédica controlada, aplicando para ello la metodología de diseñar un ventilador prototipo en condiciones de ventilador mecánico, para ser utilizado en base de sus aspectos operativos en la Clínica Internacional, siendo requisito para ello el contar con un modelo de respiración, con un modelo de operación, con una categoría de control y cuya productividad sea causal y que cuenta con alarmas si el sistema fallara, con el que concluye que se desea brindar un remedio que solvante al área de ingeniería Biomédica de la Clínica con una buena metodología y estructura.

Carrillo et al. (2022), la presente investigación forma parte de la publicación en la Revista de Ciencia y Tecnología en el Ecuador cuyo objetivo es diseñar y construir un equipo de ventilación de manera mecánica, aplicando materiales electro neumáticos con la finalidad de monitorear el ritmo cardiaco para ser suministrado de manera adecuada a los pacientes y dentro de las conclusiones efectuadas de la recopilación de datos se considera que el equipo final para aplicación debe contar con un prototipo conceptual en vista a su estructura electromecánica y para su mejor aplicación el medico quien maneja este equipo deberá de realizar mediciones de medición de manera asistida.

2.1.2 Antecedentes Nacionales

Nunura (2022), en su investigación plantea como objetivo el poder diseñar y simular un ventilador mecánico, así como el analizar los beneficios y los inconvenientes de la misma realizando una evaluación económica para la aplicación de la propuesta, en ese sentido se considera como puntos claves la fisiopatología, el diagnóstico y el tratamiento, siendo una tesis no empírica dado que se va a llevar a cabo en base a la observación de tipo aplicado bajo un estudio descriptivo cuya técnica e instrumentos a aplicar fue el análisis de documentos, la búsqueda de fichas bibliográficas y el instrumento la entrevista, cuya variable independiente es el ventilador mecánico y como variable dependiente la insuficiencia respiratoria, teniendo como indicadores de estudio las precisiones caudal desplazamiento y presiones BPM, planteándose como propuesta el diseñar un ventilador mecánico no invasivo concluyendo que se debe de calcular el torque máximo en base a la presión máxima del instrumento, así como calcular la fuerza y velocidad de la misma.

Noa y Vergara (2021), en su investigación los autores tienen como objetivo demostrar con el diseño de instalaciones eléctricas mediante la tecnología BIM y la automatización en hotel 5 estrellas, considerando como áreas de aplicación el área de comunicaciones, gestión de energía, el área de confort, el área de accesibilidad y el área de seguridad, aplicando la metodología BIM llamado también BUILDING INFORMATION MODELING en la que se relaciona y representa una estructura tridimensional y paramétrica, concluyendo que aplicando la metodología de estudio se estandariza la accesibilidad y simplificación de las 5 áreas de estudio las que no se cubren al 100% actualmente, automatizando así las instalaciones eléctricas mediante el software Revit.

2.2 Bases Teóricas

2.2.1 *El Aparato Respiratorio*

El aparato respiratorio o sistema respiratorio está formado por estructuras realizadas por la interacción entre la sangre y los gases en la atmosfera. El mecanismo que realiza el aparato respiratorio comienza cuando el oxígeno (O₂) es introducido dentro del cuerpo humano para posteriormente ser distribuido a todos los tejidos y el dióxido de carbono (CO₂) generado por el metabolismo celular es expulsado e eliminado hacia el exterior.

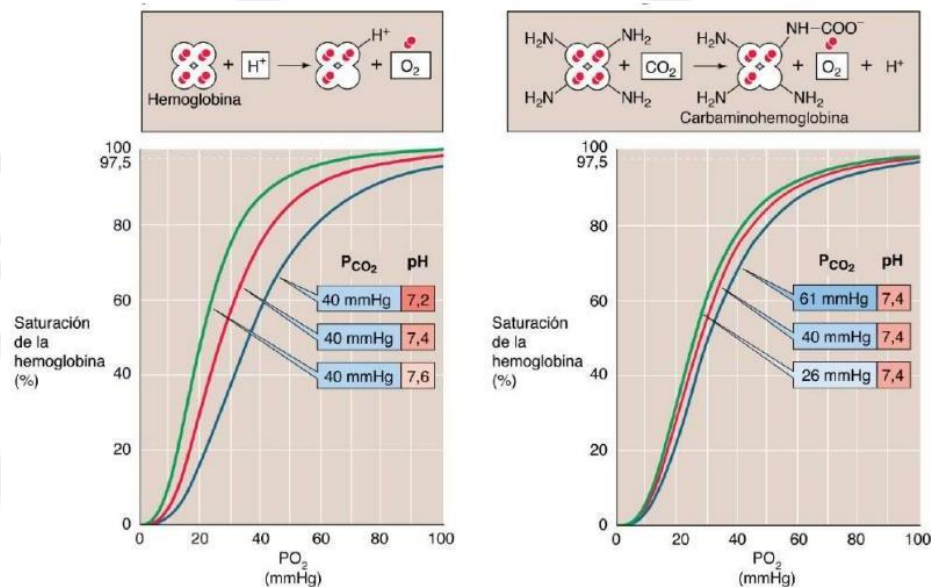
Además de estos factores también actúa la regulación pH corporal para el resguardo de agentes externos como los son los patógenos y cualquier sustancia que pueda producir efectos contraproducentes hacia el cuerpo humano.

2.2.2 *Fisiología Respiratoria y Transporte de Oxígeno*

- *Fisiología Respiratoria Infante.* Para infantes de (1-3 años), la respiración por minuto es 20-30 (considerar esto a profundidad, ya que la tesis trata de esto)
- *Fisiología Respiratoria Adulto.* Para adultos mayores a 18, la respiración por minuto es de 12-20 (considerar esto a profundidad, ya que la tesis trata de esto)
- *Transporte de Oxígeno.* El transporte comienza cuando la molécula de oxígeno (O₂) se combina de forma revocable con la dosis hemo de la hemoglobina, cuando la presión del oxígeno favorece a la unión del oxígeno (O₂) a la hemoglobina produciendo la liberación del dióxido de carbono (efecto Haldane), en cambio cuando la concentración del dióxido

de carbono (CO_2) es alta se une a la hemoglobina, reduciendo la consonancia con el oxígeno (O_2) produciendo el efecto Bohr como se observa en la Figura 3 (UNAM, 2021).

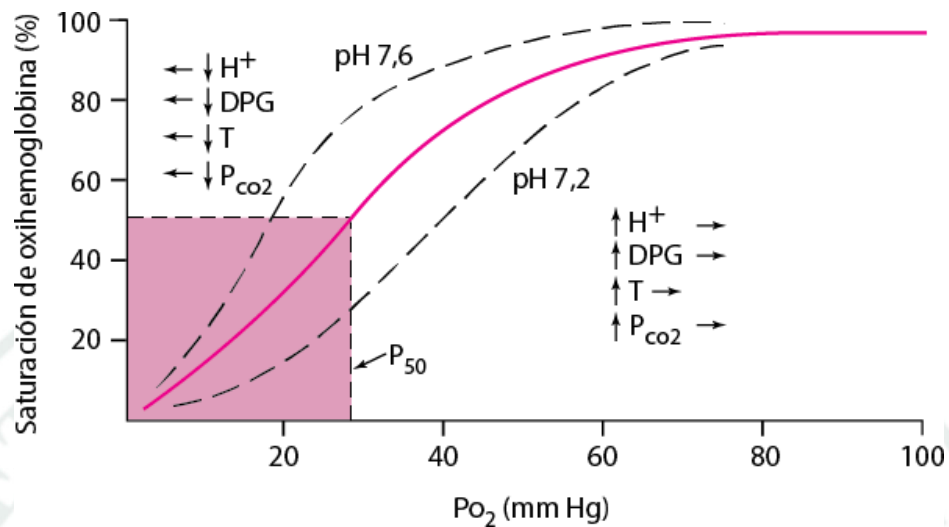
Figura 3
Efecto Bohr



Nota. Transporte de Gases (Santeliz)

El oxígeno unido a la hemoglobina (Hb) transporta el 97% , el resto se disuelve en el agua de plasma y de las células , cada porción mínima de hemoglobina puede eximir hasta 1.34 ml de oxígeno (O_2), por consiguiente, la hemoglobina de 100 ml de sangre se puede mezclar con 20 ml de oxígeno cuando la hemoglobina está saturada al 100%, a esto se le conoce como curva de descomposición de la hemoglobina (Figura 4), dicha curva muestra un incremento consecutivo del porcentaje de hemoglobina (Hb) con el oxígeno (O_2) a medida que aumenta la presión parcial de oxígeno (PO_2) en la sangre. (UNAM, 2021).

Figura 4
Curva de disociación de la hemoglobina



Nota. Manual MSD, Versión para profesionales.

2.2.3 Modalidades de Ventilación Mecánica Pulmonar

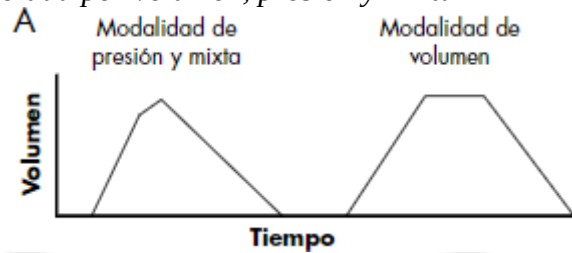
Según López y Carrillo (2008), se pueden distinguir cuatro modalidades según la programación de la ventilación mecánica y la forma de meter el aire, los cuales son:

- a) Presión negativa
- b) Presión positiva
- c) Invasivos
- d) No invasivos

Ventilación programada por volumen. El ventilador siempre entrega la misma corriente al paciente con cada respiración e inyecta aire a una velocidad constante. Es una ventilación con un volumen constante y una presión variable (Figura 5). La desventaja es que se pueden alcanzar altas presiones en las vías respiratorias, lo que genera un riesgo de barotrauma.

Figura 5

Ventilación controlada por volumen, presión y mixta



Nota. Adaptado de Ventilación mecánica: indicaciones, modalidades y programación y controles, por López y Carrillo (2008).

Ventilación programada por presión. El respirador sigue su recorrido hasta lograr la presión establecida y se mantiene firme durante todo el periodo inspiratorio con una rapidez de flujo desacelerada (Figura 6). Es una ventilación con presión constante y volumen variable. Su principal desventaja es que el volumen otorgado cambia según la disposición de la vía aérea y los pulmones del paciente, por lo que se presenta mayor riesgo de volumen trauma.

Figura 6

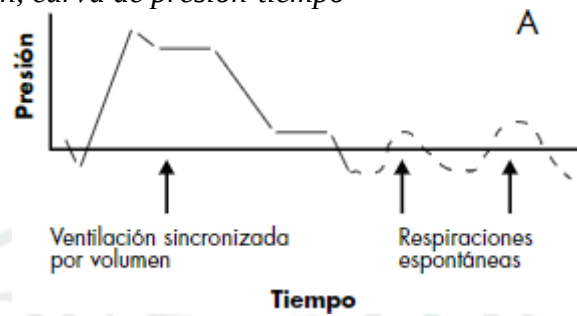
Curvas de presión-tiempo



Fuente. Adaptado de Ventilación mecánica: indicaciones, modalidades y programación y controles, por López y Carrillo (2008).

Ventilación mixta o de doble control (programada por volumen y ciclada o regulada por presión). Comparte las mismas funciones que la modalidad de volumen a la entrega la misma corriente al paciente con cada respiración y las mismas funciones que la modalidad de presión insertando el aire con una corriente ralentizada. Es una ventilación con volumen constante y presión variable (Figura 7).

Figura 7
SIMV por volumen, curva de presión-tiempo



Nota. Adaptado de Ventilación mecánica: indicaciones, modalidades y programación y controles, por López y Carrillo (2008).

2.2.4 Ventilación Mecánica

La ventilación mecánica es un mecanismo de soporte vital considerablemente empleado en entornos clínicos que involucran disfunción respiratoria de fuente intrapulmonar o extrapulmonar. Debe usarse en unidades de cuidados intensivos, pero es posible que deba usarse en servicios de ambulancia, transporte de pacientes en estado crítico y en condiciones que amenazan la vida en general.

Ventilación Mecánica Invasiva. Este tipo de ventilación es un procedimiento que enlaza un ventilador al paciente mediante un tubo endotraqueal o traqueotomía para reemplazar la función ventilatoria. (Martínez, 2019).

Ventilación Mecánica No Invasiva. Es una importante alternativa terapéutica cuando se debe evitar la ventilación invasiva. Brinda algún beneficio al paciente, pero tiene instrucciones particulares y puede requerir que se coloquen accesorios entre el ventilador y el paciente (mascarilla nasal o facial, ventilaciones) con el fin de asegurar los beneficios. Se asocia primordialmente en la insuficiencia respiratoria de la hipercapnia o hipoxémica, defecto de respiración aguda, pacientes que aguardan un trasplante pulmonar y pacientes que no son aspirantes a intubación: (causas reversibles de LRA) como el edema pulmonar cardiogénico sin fluctuación hemodinámica, defecto en el

postoperatorio pulmonar, insuficiencia respiratoria en pacientes con SIDA y EPOC.
(Urrutia y Cristancho, 2006)

2.2.5 Medición de Flujo y Presión de Aire (en tubos)

La medición de flujo se realiza para determinar las dimensiones en masa o en volumen de los fluidos insertados en un proceso y disponer la medida de fluido empleado por el proceso con el objetivo de controlar y calcular los costos.

La medición de la presión se basa en aplicar la ley de conservación de la energía al flujo midiendo el diferendo de presión entre dos puntos de flujo diferentes. Este cambio de velocidad es causado por una disminución en el área (placa de orificio, boquilla de flujo) o una disminución en la velocidad a cero (tubo de Pitot, tubo anular). (Dulhoste, 2008)

2.2.6 Teorema de Bernoulli

El teorema de Bernoulli aplicado entre un punto en la tubería y un punto en la contracción es el siguiente.

$$\frac{\rho_1}{\rho_1 g} = \frac{V_1^2}{2g} + z_1 = \frac{\rho_2}{\rho_2 g} = \frac{V_2^2}{2g} + z_2 \quad (1)$$

Usando la ecuación de la conservación de la masa (continuidad)

$$q_{m1} = q_{m2}; \quad \rho_1 V_1 A_1 = \rho_2 V_2 A_2$$

$$z_1 = z_2 \quad y \quad \rho_1 = \rho_2 = \rho$$

La ecuación de Bernoulli queda:

$$\frac{p_1}{\rho} = \frac{V_1^2}{2} = \frac{p_2}{\rho} = \frac{V_2^2}{2}$$

Reordenando

$$\frac{p_1 - p_2}{\rho} = \frac{V_2^2 - V_1^2}{2}$$

Ecuación de continuidad

$$V_1 = V_2 \frac{A_2}{A_1} \quad (2)$$

Donde:

$$A_1 = \frac{\pi D^2}{4} = \frac{\pi d^2}{4\beta^2}; \quad A_2 = \frac{\pi d^2}{4}; \quad p_1 - p_2 = \Delta p$$

$$\beta = \frac{d}{D} \quad \text{Relación de diámetro}$$

D = Diámetro de tubería

d = Diámetro de contracción

Obtenemos:

$$V_2^2 - \left(V_2 \frac{A_2}{A_1}\right)^2 = 2 \frac{\Delta p}{\rho}; \quad V_2^2 \left(1 - \left(\frac{d^2}{D^2}\right)^2\right) = 2 \frac{\Delta p}{\rho}; \quad V_2^2 \left(1 - \left(\frac{\beta^2 d^2}{d^2}\right)^2\right) = 2 \frac{\Delta p}{\rho};$$

$$V_2^2 (1 - \beta^4) = 2 \frac{\Delta p}{\rho}; \quad V_2 = \frac{1}{\sqrt{(1 - \beta^4)}} \sqrt{2 \frac{\Delta p}{\rho}}; \quad q_v = V_1 A_1 = V_2 A_2 = V_2 \frac{\pi d^2}{4}$$

La ecuación básica para medición de flujo es:

$$q_v = \frac{1}{\sqrt{(1-\beta^4)}} \frac{\pi d^2}{4} \sqrt{2 \frac{\Delta p}{\rho}} \quad (3)$$

2.2.7 Sensores, reguladores de aire y oxígeno

Sensor de presión de aire

Los transductores de presión piezorresistivos diferencial de la serie MPxx5010 son sensores de presión de silicio de última generación diseñados para una amplia gama de aplicaciones, pero particularmente para aquellas que emplean un microcontrolador o microprocesador con entradas A/D. Este transductor combina técnicas avanzadas para proporcionar una señal de salida analógica precisa que es proporcional a la presión aplicada, trabaja con presiones en el rango de 0 a 115 kPa.

Figura 8
Sensor de presión de aire



Nota. Superhitv https://superhitvs.live/product_details/9192446.html

Sensor de flujo de aire

El sensor de flujo de aire tiene la función de determinar la cantidad precisa de aire que entra al motor mediante el colector de admisión; a su vez esta contiene una placa metálica que proporciona tensión confiable para mitigar problemas de ralentización irregular del motor, también la placa de circuito impreso de cerámica que brinda mayor resistencia a los cambios de temperatura y vibraciones. (Spectra Premiun)

Figura 9
Sensor de flujo de aire



Nota. Spectra Premium <https://www.spectrapremium.com/es/aftermarket/north-america/mass-air-flow-sensor>

Regulador de oxígeno

Los reguladores de oxígeno son mecanismos que disminuyen la presión del oxígeno desde un cilindro hasta un nivel seguro para su uso, en caso de fuga se activa si sistema de seguridad que alivia el exceso de presión en caso de presentarse una fuga de manera imprevista. (MD dispositivos médicos del Perú)

Figura 10
Regulador de oxígeno



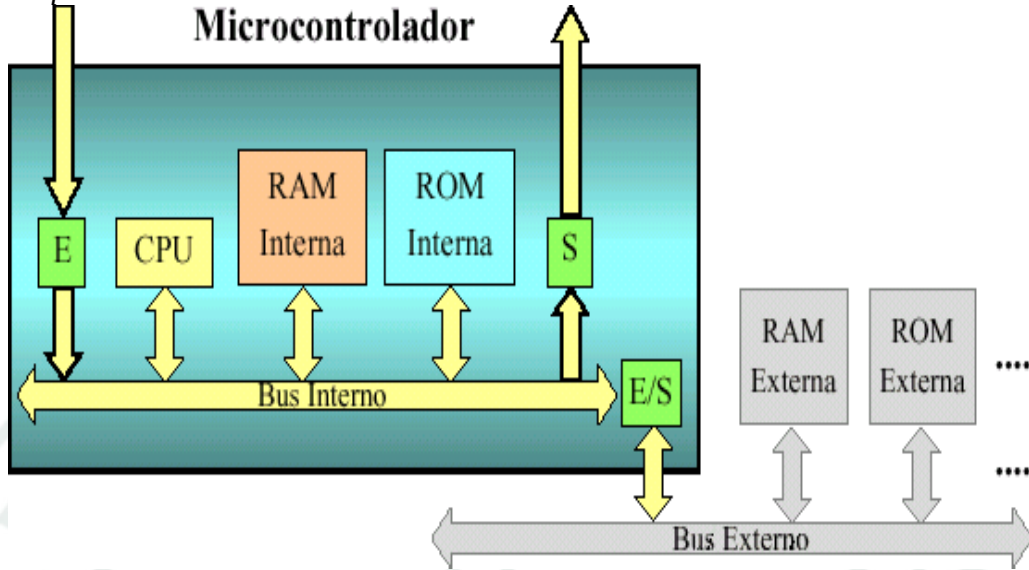
Nota. <https://www.promart.pe/regulador-de-oxigeno-1000182792/p>

2.2.8 Microcontroladores

Un microcontrolador es una red compuesto y consistente, programable que posee los implementos imprescindibles para realizar y verificar una tarea específica. El número de componentes integrados en un microcontrolador varía según el diseño del fabricante, pero los implementos fundamentales que suelen ser microprocesadores, memoria RAM, memoria de programa, convertidores A/D, osciladores, puertos de comunicación, etc. (Figura 11). Esto le dio a este tipo de dispositivo una gran versatilidad, y hoy en día su uso se ha incrementado significativamente en todo el mundo. (Palacios et al., 2004).

Figura 11

Arquitectura del microcontrolador



Nota. Adaptado de Microcontrolador vs PLC (Electrotec, 2015). Tomado de <https://electrotec.pe/blog/microcontroladorvsplc>

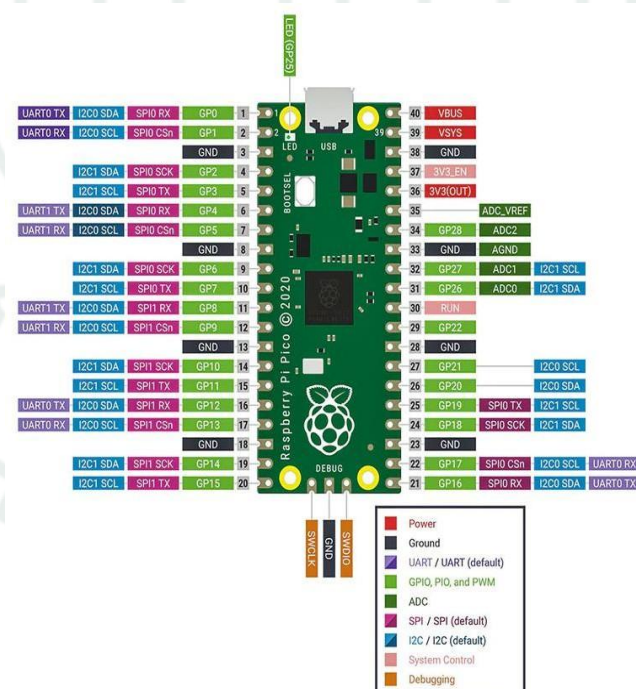
Tipos y familias de microcontroladores

Entre las principales tipos y familias tenemos:

- Familia AVR: Desarrollada por la empresa Atmel, esta familia de microcontroladores es muy utilizada en proyectos de electrónica y robótica debido a su bajo consumo de energía y gran cantidad de pines de entrada y salida.
- Familia PIC: Desarrollada por la empresa Microchip, esta familia de microcontroladores es muy popular en la industria debido a su bajo costo y gran cantidad de periféricos integrados.
- Familia ARM: Desarrollada por la empresa ARM Holdings, esta familia de microcontroladores es muy utilizada en proyectos de alta complejidad debido a su gran competencia de procesamiento y bajo consumo de energía.

- Familia MSP430: Desarrollada por la empresa Texas Instruments, esta familia de microcontroladores se resalta por consumir poca energía y gran cantidad de periféricos integrados.
- Raspberry Pi Pico: Es un microcontrolador de placa única desarrollado por Raspberry Pi Foundation. Sin embargo, las computadoras Raspberry Pi tradicionales, el Pico está diseñado específicamente para aplicaciones de microcontrolador y programación de bajo nivel por medio del chip RP2040. Esta herramienta se caracteriza por presentar un microcontrolador RP2040, dos núcleos ARM Cortex M0 con 264KB de memoria SRAM y 2MB de almacenamiento en memoria Flash, 26 pines GPIO multifunción, interfaces SPI, I2C y UART, sensor de temperatura y también porque se puede programar en C y MicroPython.

Figura 12
Diagrama de pines



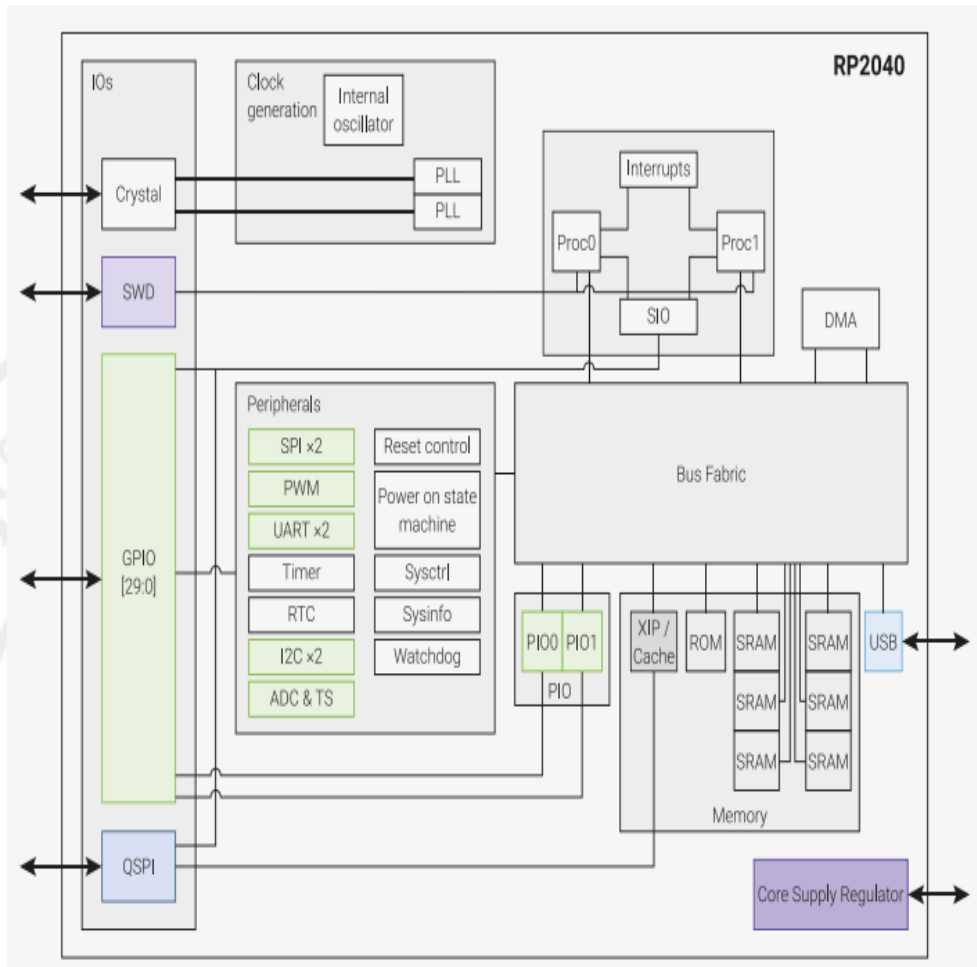
Nota. Robotopia <https://robotopia.es/img/cms/raspberry-pi-pico- diagrama-de-pines.jpg>

Microcontrolador RP2040. El RP2040 es el primer microcontrolador de Raspberry Pi. Aporta valores característicos de alto rendimiento, bajo coste y facilidad de uso al espacio de los microcontroladores. Con una gran memoria en el chip, un complejo procesador simétrico de doble núcleo, una estructura de bus determinista y un rico conjunto de periféricos y un amplio conjunto de periféricos, que se complementa con nuestro exclusivo subsistema de E/S programable (PIO), ofrece a los usuarios profesionales una potencia y una flexibilidad inigualables.

Con una documentación detallada, un puerto Micro Python pulido y un cargador de arranque UF2 en ROM, tiene la barrera de entrada más baja posible para los usuarios principiantes y aficionados.

El RP2040 es un dispositivo sin estado, con soporte para la ejecución en caché desde la memoria QSPI externa (Figura 13). Esta forma de diseño le permite elegir la densidad adecuada de almacenamiento no volátil para su aplicación, y beneficiarse del bajo precio de las piezas Flash de uso corriente. (RP2014 DATASHEET, 2022)

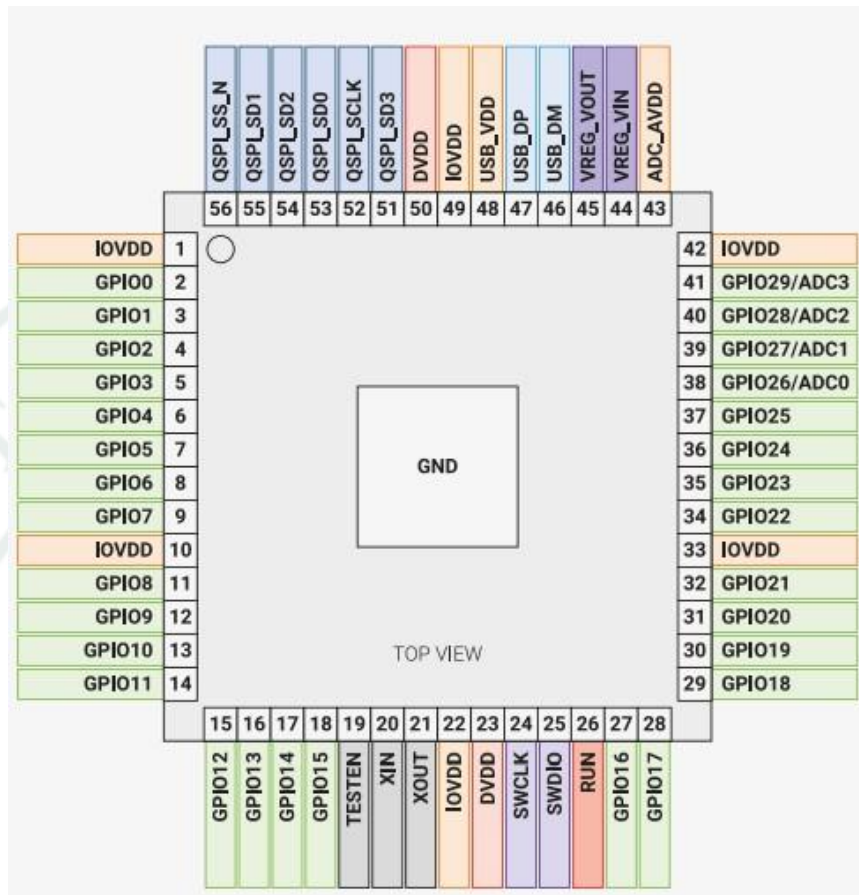
Figura 13
Chip del sistema RP2040



Nota. RP2040 DATASHEET – A microcontroller by Raspberry Pi, por RP2014 DATASHEET, (2022).

El RP2040 cuenta con dos núcleos de procesador M0+, DMA, memoria interna y bloques periféricos conectados a través de la estructura del bus AHB/APB de tejido (Figura 13).

Figura 14
Ubicación de los pines



Nota. RP2040 DATASHEET A microcontroller by raspberry Pi, por RP2014 DATASHEET, (2022).

2.2.9 Lenguajes de Programación

Existen varios lenguajes de programación que se utilizan comúnmente para programar microcontroladores. Algunos de los lenguajes más populares son:

- C: Es uno de los lenguajes más utilizados para programar microcontroladores debido a su eficiencia y capacidad de acceso directo a los recursos del hardware.
- Assembly: Es un lenguaje de bajo nivel que permite un control preciso sobre el hardware del microcontrolador. Se utiliza especialmente en

aplicaciones que requieren un alto rendimiento y una programación muy específica.

- **Arduino:** Aunque técnicamente no es un lenguaje de programación en sí, Arduino utiliza un dialecto de C/C++ simplificado para programar sus microcontroladores. Es muy popular entre los aficionados y principiantes debido a su fácil aprendizaje y amplia comunidad de soporte.
- **MicroPython:** Es una implementación del lenguaje de programación Python optimizada para microcontroladores. Proporciona una sintaxis más sencilla y legible en comparación con otros lenguajes de bajo nivel.

Micropython, es una adquisición elemental y eficaz del lenguaje de programación Python 3 que inserta un pequeño subconjunto de la biblioteca estándar de Python y está perfeccionado para el desarrollo en microcontroladores y en entornos limitados. MicroPython está repleto de funciones superiores, como un instructivo participativo, enteros de precisión arbitraria, cierres, alcance de listas, procesadores, conducción de excepciones y más. Sin embargo, es convenientemente compacto como para contener y funcionar con solo 256k de espacio de código y 16k de RAM. (George, 2014)

Se sabe además que MicroPython es una versión nueva que viene adaptada de Python, la cual es un lenguaje descifrable recopilado para efectuarse, sino que se infiere durante el proceso. Gracias a este tipo de lenguaje se pueden cumplir diversos propósitos comprobando las entrada y salida de un microcontrolador, mediante las señales digitales y analógicas. (Rivera, 2022)

Para programar interfaces gráficas, hay varios lenguajes de programación que se utilizan comúnmente, como:

- Java: Es ampliamente utilizado para desarrollar interfaces gráficas de usuario (GUI) debido a su portabilidad y a la disponibilidad de bibliotecas como Swing y JavaFX.
- Python: Es un lenguaje popular para la creación de interfaces gráficas debido a su simplicidad y a la disponibilidad de bibliotecas como PyQt y Tkinter.
- C#: Es un lenguaje de programación desarrollado por Microsoft y se utiliza principalmente para el desarrollo de aplicaciones en el entorno de Windows. C# ofrece herramientas y bibliotecas como Windows Forms y WPF para crear interfaces gráficas.
- JavaScript: Aunque es más conocido por el desarrollo web, JavaScript también se utiliza para crear interfaces gráficas interactivas en aplicaciones de escritorio y para web mediante bibliotecas como React y Vue.js.
- Visual Basic: Es un lenguaje de programación desarrollado por Microsoft y se utiliza principalmente para el desarrollo de aplicaciones en el entorno de Windows. Visual Basic ofrece su propio entorno de desarrollo integrado (IDE) y herramientas para crear interfaces gráficas utilizando Windows Forms.

Visual Basic, Es un lenguaje de programación que se centra en objetos, es decir, permite crear aplicaciones .NET o funciones de forma rápida y sencilla; siendo este una estrategia de lenguaje con un modelo seguro, ya que también es asequible.

Una aplicación de Visual Basic está compuesta por bloques de innovación estándar. La versión actual es la 16.9/Visual Studio 2019, es decir, la versión 16.9, la cual busca proveer analizadores, correcciones de código y cualidades de eficiencia del entorno de desarrollo integrado (IDE).

Por ello, Visual Basic es uno de los lenguajes más empelados para la red operativa de Microsoft Windows, con la capacidad de incluir a más lenguajes en un mismo entorno, como son el lenguaje de Visual C# y el lenguaje Visual C++.

2.2.10 Entorno de Desarrollo (IDE)

Microsoft Visual Studio. Visual Studio (Figura 15) es una agrupación integra de herramientas de desarrollo para desencadenar aplicaciones web ASP.NET, Servicios Web XML, programas y/o aplicaciones de escritorio y móviles. Visual Basic, Visual C# y Visual C++ emplean todo el mismo entorno de desarrollo integrado (IDE), que acredita el empleo compartido de implementos y posibilita la creación de alternativas que resuelven en varios lenguajes. Por ello, estos lenguajes emplean las tareas de .NET Framework, las cuales otorgan la entrada a tecnologías clave para facilitar el desarrollo de aplicaciones y/o programas web ASP y Servicios Web XML. (Learn Microsoft, 2022).

Figura 15
Logotipo Visual Studio

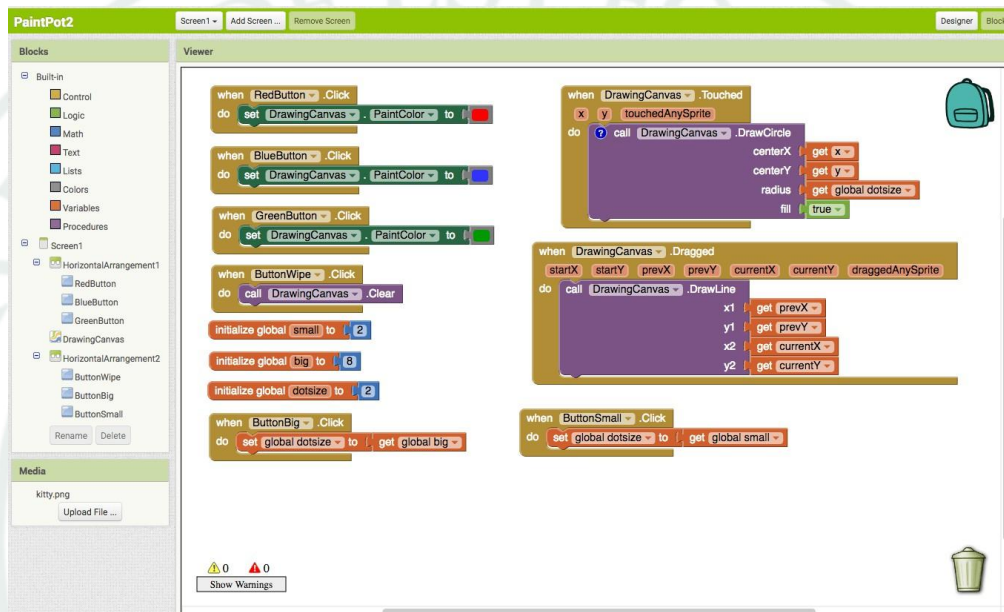


Nota. MVP Cluster, por Learn Microsoft, (2022).

App Inventor 2, App Inventor es un entorno de desarrollo visual de aplicaciones para dispositivos Android. Para desarrollar aplicaciones con App Inventor es necesario únicamente un navegador web y un teléfono o tableta Android. App Inventor se basa en

un encargo web que te autoriza guardar tus tareas facilitando el control y seguimiento de estas tareas o proyectos (Figura 16). App Inventor 2 es también un entorno de estudio de lenguaje de programación sencillo de usar, con la que incluso los no programadores pueden ejecutar sus propias aplicaciones. (Gómez y Larraz, 2016)

Figura 16
Interfaz en App Inventor

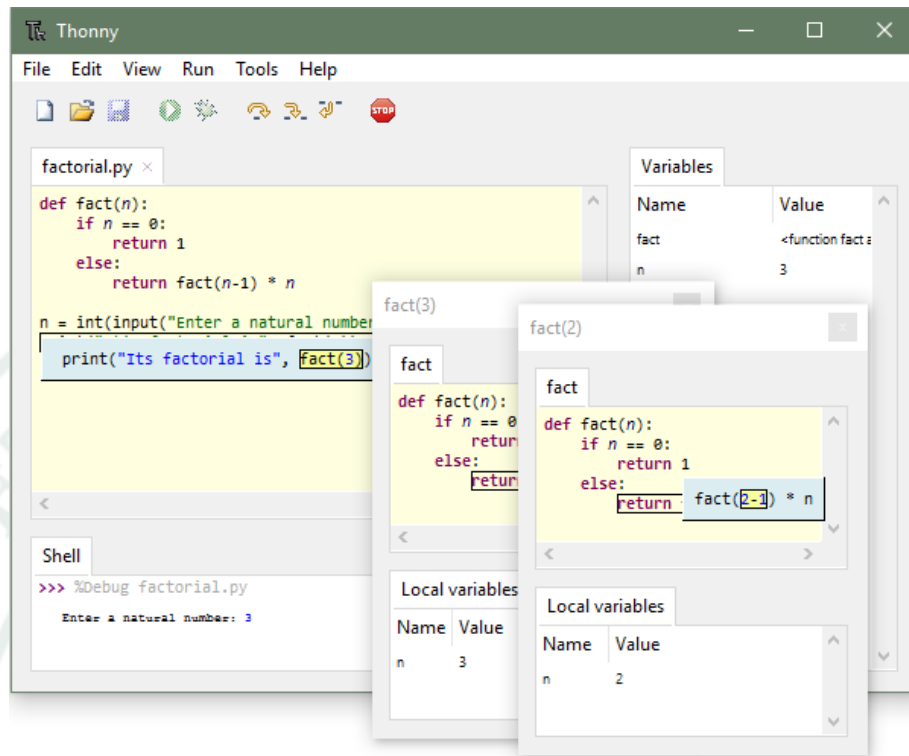


Nota. Guía rápida de App Inventor por Gómez y Larraz, (2016).

Thonny. Es un programa fundamental para poder continuar con Python, ya que trabaja con tres de las herramientas bases para trabajar con el lenguaje: intérprete, editor y depurador (Figura 17).

Thonny viene con Python 3.10 incorporado, por lo que se requiere únicamente de un instalador simple y está listo para aprender a programar; también es posible utilizar una instalación separada de Python. La interfaz de usuario inicial esta carente de todas las funciones que pueden distraer a los que están iniciando a aprender. (University of Tartu, 2018)

Figura 17
Thonny



Nota. Thonny Python IDE for beginner, por University of Tartu, (2018)

2.3 Marco Conceptual Biológico

2.3.1 *Asimetrías de Ciclos de Inhalación y Ciclos de Respiración*

El proceso de la respiración es un ciclo que consiste en intercambiar gases; mediante las fases de inspiración, y la expiración, donde ingresa el oxígeno y eliminamos dióxido de carbono.

La fase de expiración es sutilmente más larga que la fase de inspiración; pero existen algunas alteraciones que hacen variar el periodo o tiempo de las fases. Normalmente la frecuencia respiratoria en un estado de reposo es de 12 a 15 respiraciones por minuto.

Taquipnea: Es la respiración acelerada que se produce por esfuerzo físico o la expresión de emociones, también se produce en enfermedades pulmonares o cardíacas.

Bradipnea: Es la respiración desacelerada que se provoca por la intoxicación con medicamentos, o por enfermedades del sistema nervioso.

Hiperpnea: Es la respiración profunda que presentan personas que se encuentran enfermas de acidosis metabólica.

Hipopnea: Es la respiración disminuida, es decir deficiencia respiratoria, producida por la debilidad de los músculos respiratorios.

Respiración de Biot: Esta respiración es rápida e irregular, ya que presenta periodos de apnea que suelen ser de 10 a 30 segundos.

Respiración de Cheune – Stokes: Esta respiración presenta movimientos respiratorios frecuentes y profundos en un periodo, pasando a otro periodo donde la respiración es lenta, con presencia de apnea.

Respiración suspirosa. Es aquella que presenta interrupciones por inspiraciones profundas.

2.3.2 Frecuencia Respiratoria

Este es el número de respiraciones por minuto. Puede medir su frecuencia respiratoria contando la cantidad de veces que inhala en un minuto mientras está parado (Bohórquez et al., 2002).

Método de medición preciso:

- Siéntate y trata de relajarte.
- Lo mejor es medir su frecuencia respiratoria mientras está sentado en una silla o en la cama.

- Mida su frecuencia respiratoria contando cuántas veces se infla su pecho o abdomen en un minuto.
- Guardar este número.

La frecuencia respiratoria normal para adultos en reposo es de 8 a 16 respiraciones por minuto, mientras que, para los bebés, la frecuencia respiratoria normal es de hasta 44 respiraciones por minuto. La respiración es el proceso de intercambiar sustancias por oxígeno y exhalar dióxido de carbono. Está controlado por un sistema corporal llamado actividad respiratoria.

Las operaciones de respiración se dividen en tres sistemas (Aya Perlaza):

Control nervioso central. El sistema nervioso central determina la tasa de ventilación y la cantidad de aire que se aspira. Esto afecta la naturaleza de la inhalación y exhalación (Terry y Weaver, 2012).

Sistema de entrada sensorial. El sistema sensorial le permite al sistema nervioso central saber cuánto y qué tan rápido respira. También reconoce los cambios químicos como desencadenantes (Dehaene, 2019).

Sistema muscular. El sistema muscular hace que los pulmones se muevan de acuerdo con las señales de otros sistemas. Regula el mecanismo respiratorio.

2.3.3 *Humidificador*

Un humidificador es un dispositivo que emite niebla para aumentar la humedad en el aire. Los tipos de humidificadores incluyen: Humidificadores centrales. Estos dispositivos están integrados con el sistema de calefacción y aire acondicionado en el hogar (Tocagón, 2015).

Hacemos hincapié en que los humidificadores solo se utilizan con una humedad relativa baja, por debajo del 40%. No debe estar encendido todo el tiempo para lograr el nivel de humedad relativa adecuado (Cardona et al., 2014).

Los humidificadores pueden ayudar a las personas que experimentan:

- Piel seca
- Ojos irritados
- Garganta seca o vías respiratorias
- Alergia
- Toser a menudo
- Epistaxis
- Sinusitis
- Dolor de cabeza
- Labios agrietados

2.3.4 Presión Pico

Esta es la presión obtenida inmediatamente después del final del volumen corriente (VT) de soplado. Es equivalente a la presión requerida para vencer la resistencia al flujo por fricción de las vías respiratorias y los tubos endotraqueales y la resistencia elástica del sistema respiratorio. Presión de pico (Ppk) depende de muchos factores, muchos de los cuales no están relacionados con el volumen alveolar, por lo que no es un buen indicador del estiramiento alveolar (Belda, 2009) .

Factores que afectan a presión de pico (Ppk):

A) Derivado de tubo endotraqueal: tamaño pequeño; Doblar; asignación.

B) Aumento de la resistencia de las vías respiratorias: broncoespasmo; enfermedad pulmonar obstructiva crónica; asignación.

C) Estiramiento alveolar.

D) Alto flujo inspiratorio.

E) Aumento de la presión pleural o transtorácica: tos; neumotórax; ascitis.

Todos estos factores determinan el desarrollo de presión de pico (Ppk). Aunque experimentalmente a valores > 50 cm H₂O. Con pérdida de integridad alveolar, presión de pico (Ppk) no es un buen predictor del riesgo de barotrauma porque mide tanto la resistencia elástica (expansión pulmonar) como la resistencia friccional (vías respiratorias, tubo endotraqueal, tráquea). La diferencia entre la presión máxima medida en las vías respiratorias y la presión máxima medida en la tráquea representa la resistencia al flujo del tubo endotraqueal (Belda, 2009).

2.3.5 Presión Residual

Corresponde al volumen de aire que queda en los pulmones después de la exhalación máxima (Oblitas, 2008).

La forma más precisa es la medición del volumen corporal. Estás sentado en una caja de vacío transparente que parece una cabina telefónica. El técnico le pide que inhale y exhale en la boquilla. Los cambios de presión dentro de la caja ayudan a determinar la capacidad pulmonar.

El volumen corporal (BC) es una prueba de función respiratoria que mide la capacidad residual funcional (FRCpleth) y la resistencia específica de las vías respiratorias (sRaw). PC es el estándar de oro para la medición del volumen pulmonar (Guerrero et al., 2016).

2.3.6 Volumen Tidal

Este es el volumen de aire que entra y sale de los pulmones durante la respiración nasal. Volumen de reserva inspiratorio (IRV) también es el volumen de aire que circula entre una inhalación normal y una exhalación sin ningún esfuerzo extra. Un valor normal es de unos 500 ml o 7 ml/kg de peso corporal. Para calcular el volumen corriente, siga estos pasos (Hulbert, 2009):

1. Medir la altura del paciente.
2. Calcula el peso = (altura (en metros) x altura) x 23 para hombres o 21,5 para mujeres.
3. Multiplica el peso ideal de 6 a 10 ml (PROMEDIO 8 ml).
4. Compruebe inmediatamente la presión máxima <35 cmH2O.

2.4 Marco Conceptual Tecnológico

2.4.1 Fritzing

Fritzing es un programa de automatización de diseño electrónico gratuito diseñado para ayudar a los diseñadores y artistas a pasar de un prototipo (por ejemplo, usando una placa de prueba) al producto final (Abad Carpintero, 2015).

Fritzing se creó siguiendo Arduino y los principios de procesamiento y permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear diagramas de circuitos impresos para agregarlos a la producción. Además, tiene un sitio web complementario para ayudar a compartir y discutir bocetos y experiencias y reducir los costos de producción (Luengo, 2020).

Fritzing se puede considerar como un programa de automatización de diseño electrónico (EDA) para personas que no son ingenieros. A partir de 2014, Fritzing también tiene una vista de código fuente para escribir y programar dispositivos Arduino directamente (Copa, 2017).

2.4.2 APP

El término "aplicación" es un acrónimo de la palabra en inglés application y se usa comúnmente para referirse a las aplicaciones de escritorio para dispositivos móviles y tabletas.

Aplicación nativa. Una aplicación nativa es una aplicación diseñada específicamente para un sistema operativo en particular, conocida como kit de desarrollo o SDK. Cada plataforma, Android, iOS o Windows Phone, tiene su propio sistema operativo, por lo que, si quieres que tu app esté disponible en todas las plataformas, tendrás que crear algunas apps en el idioma del sistema operativo que hayas seleccionado (Avila et al., 2016).

Aplicación Web. Es una versión del sitio web optimizada y adaptada a todos los dispositivos móviles. En otras palabras, es una página que se puede abrir desde el navegador de cualquier terminal, independientemente del sistema operativo utilizado. Esta optimización es posible gracias a HTML5 y CSS3. Las aplicaciones web se ejecutan en el navegador web de su dispositivo a través de una URL. Por ejemplo, en Safari si es plataforma iOS. El contenido se adapta a la pantalla, tiene apariencia de APP de navegación (Avila et al., 2016; Robledo, 2016).

Aplicación híbrida (o aplicación web nativa). una aplicación híbrida que se encuentra entre una aplicación nativa y una aplicación web. Se desarrolla como

aplicaciones web, es decir, utilizando HTML5, JavaScript y CSS3. Además, las herramientas móviles como la cámara o el GPS están disponibles sin restricciones, al igual que en las aplicaciones nativas (Godoy et al., 2022).

Aplicaciones Web Progresivas. estas aplicaciones son una creación muy nueva. En pocas palabras, estos son sitios web que se comportan como aplicaciones nativas. Desde un punto de vista más técnico, se podría decir que "es un término para la próxima generación de aplicaciones que aumentan su funcionalidad a medida que aumentan las capacidades del dispositivo en el que se ejecutan, de ahí la palabra" acumulativamente " (Fumero et al., 2007).

2.4.3 SolidWorks

SolidWorks es un software de modelado mecánico (CAD) 2D y 3D desarrollado actualmente por SolidWorks Corp., una subsidiaria de Dassault Systèmes (Suresnes, Francia), para el sistema operativo Microsoft Windows (Álvarez, 2017).

SolidWorks es un software de diseño mecánico similar a CAD que utiliza un entorno gráfico basado en Microsoft Windows para crear modelos, ensamblajes y dibujos en 3D de manera intuitiva y rápida. El trabajo de SOLIDWORKS en el desarrollo de productos es muy específico, las soluciones ayudan a acelerar el proceso, ahorran tiempo y dinero y dan cabida a la innovación de productos (Álvarez, 2017) .

Si bien la cadena de valor de la mayoría de las empresas es un proceso secuencial en el que deben completar etapas anteriores para comenzar otras nuevas, las soluciones de SOLIDWORKS permiten que el proceso se ejecute en paralelo en lugar de secuencialmente, lo que ahorra tiempo y permite tomar mejores decisiones comerciales. Crea los mejores proyectos (Morrás, 2020) .

CAPITULO III

METODOLOGIA Y DIMENSIONAMIENTO DEL PROTOTIPO

En el presente capítulo se presenta la metodología a seguir para obtener el diseño y selección de los componentes del prototipo de ventilador mecánico con operación manual y remoto para conseguir un ventilador mecánico que satisfaga las necesidades del usuario reflejándose en sus características y especificaciones técnicas.

3.1 Metodología de Diseño

En primer lugar, se realizó un esquema del orden en el que se desarrolló el diseño conceptual representado en la Figura 18:

Figura 18
Mapa conceptual de la metodología de diseño



Nota. Elaboración propia.

3.1.1 Requerimientos del Usuario

Se identificaron las necesidades básicas que deben ser satisfechas por el diseño del ventilador mecánico basado en AMBU, a partir de una consulta realizada a profesionales de la salud en la Región de Arequipa. Se tomaron en cuenta las sugerencias de estos usuarios de ventiladores mecánicos y se elaboró una lista de características que

esperan que el dispositivo diseñado tenga. Estas características, denominadas requerimientos del usuario en este proyecto, están relacionadas con los objetivos establecidos para el diseño y las necesidades básicas. La Tabla 1 muestra un resumen de estos requerimientos y se pueden encontrar más detalles de esta evaluación en el anexo H, donde se incluyen las opiniones de médicos, enfermeros y pacientes de los hospitales de la ciudad de Arequipa.

Tabla 1

Cuadro de las características más importantes en los ventiladores mecánicos desde el punto de vista del usuario.

Características	Escala				
	1	2	3	4	5
Diseño compacto				X	
Transportabilidad					X
Calibración				X	
Tiempo de vida					X
Seguridad del paciente					X
Funcionamiento				X	
Monitoreo					X
Señales de emergencia					X
Seguridad					X
Ruido					X
Costo operativo			X		
Costo de mantenimiento				X	

Nota. Elaboración propia.

La 1, pertenece a la recopilación de sugerencias por profesionales de salud de UCI, donde se determinó emplear una escala que va del número 1 (siendo la de menor importancia) hasta el número 5 (siendo la de mayor importancia), esto con el objetivo de conseguir información para alcanzar los aspectos más importantes que los profesionales

de la salud consideran que debería tener un ventilador mecánico basado en bolsa AMBU. Luego se seleccionó y ordenó de forma más técnica: Fácil transferibilidad, alta durabilidad, fácil monitoreo, sistema de alerta óptimo, seguridad para el operador, seguridad para el paciente, bajos niveles de ruido, diseño compacto, fácil calibración, asequible, costo operativo rentable, bajo costo de mantenimiento, siendo ahora conocidas como requerimientos del usuario.

3.1.2 Punto de Vista de Fabricación

En el apartado mencionado, se consultó a expertos en diseño y fabricación de máquinas de ingeniería mecánica, mecatrónica y electrónica de la Universidad Católica de Santa María. Estos expertos sugirieron agregar características desde el punto de vista de fabricación, como el uso de materiales locales, facilidad de ensamblaje, disponibilidad de repuestos, bajo costo de manufactura y bajo consumo energético. Estas características ahora se conocen como requerimientos funcionales. Para obtener información y sugerencias al respecto, se realizó una encuesta a 25 personas de diferentes áreas, la cual se adjunta en el anexo H.

La Tabla 2 fue elaborada utilizando los datos de la encuesta. En esta tabla se establece la relación entre los requisitos funcionales y las demandas del cliente, mediante una técnica conocida como Despliegue de la Función de Calidad (QFD). Esta metodología nos permite vincular la información proporcionada por el usuario con las metas de diseño establecidas, con el fin de lograr un producto de alta calidad.

En esencia, se obtuvo un orden de prioridad de los requisitos funcionales, lo que orientó el diseño en función de esta priorización, tal como se detalla en la Tabla 2. La ponderación de la columna 9 se logra sumando los valores de la columna 3 a la 8 y multiplicando con la primera columna. Esta suma representa la puntuación técnica para

cada requisito funcional. Luego, se compara este resultado con una importancia del 100% para cada requisito funcional. En conclusión, se debe de dar mayor importancia al costo de producción, seguido por la necesidad de que el ventilador mecánico cumpla con parámetros de medidas de presión y volumen. Asimismo, se considera significativo el tamaño de construcción del ventilador mecánico, mientras que se otorga menor importancia al consumo energético.

Tabla 2

Cuadro de relación entre requerimientos del usuario y requisitos funcionales

1: bajo, 5: alto	Requisitos Funcionales ->	Consumo energético	Costo de producción	Fácil ensamble	Mantenimiento	Medición de presión y volumen	Tamaño	Puntuación ponderada
Calificación por Usuario	Requisitos del Usuario							
5	Facilidad de Transporte	3	5	5	3	1	5	110
5	Duración larga	5	5	3	3	3	3	110
5	Monitoreo básico	3	5	3	3	5	5	120
5	Sistema de alerta	3	5	3	3	5	3	110
5	Seguridad del operador	3	5	3	5	5	3	120
5	Seguridad del paciente	3	3	3	5	5	5	120
5	Bajo nivel de ruido	1	5	3	3	5	5	110
4	Diseño compacto	3	5	5	3	3	5	96
4	Calibración sencilla	3	3	5	3	5	3	88
4	Facilidad de uso	1	3	3	3	5	5	80
3	Bajo costo operativo	5	3	3	5	5	3	72
4	Bajo costo de mantenimiento	3	5	3	5	3	1	80
	Puntuación de importancia técnica	160	238	188	196	224	210	1216
	Importancia %	13.16%	19.57%	15.46%	16.12%	18.42%	17.27%	100.00%
	Rango de prioridades	6	1	5	4	2	3	
Relación 1 (débil), 3 (medio), 5 (fuerte)								

Nota. Elaboración propia.

En función a los requisitos del usuario, los fines de diseño determinados para el proyecto, las limitaciones descritas de manera general, y la data médica detallada en el marco teórico, se procedió a decidir los requerimientos de diseño de la maquina o dispositivo, por ello, primero se ejecutó una Tabla en cuya columna izquierda detalla los requerimientos del usuario y los requisitos funcionales establecidos por los diseñadores,

con su correspondiente medio para satisfacer dicho requerimiento. Las características descritas como medio para satisfacer los requerimientos (Tabla 3) fueron la base para proponer las alternativas de solución al problema planteado inicialmente, este paso fue fundamental, ya que así nos aseguramos de cumplir con los objetivos que se establecieron anteriormente y que estos guarden la mayor relación con lo que quiere el cliente/usuario del ventilador mecánico basado en resucitador AMBU.

Tabla 3

Medios para satisfacer los requerimientos planteados para el diseño

Requerimientos	Medio para satisfacer los requerimientos
Fácil Transportabilidad	Material ligero, acrílicos, PVC, cantidad de piezas reducidas, tamaño no mayor a 70 x 50 x 30 cm, construcción simple, ensamble modular
Alta durabilidad	Materiales resistentes, PVC, acero, recubrimiento de alta Material ABC.
Monitoreo Sencillo	Sensores analógicos, Raspberry Pi Pico
Seguridad para el Operador	Aislamiento de partes energizadas
Seguridad para el Paciente	Sensores analógicos, Raspberry Pi Pico, materiales resistentes
Bajos Niveles de Ruido	Motor de tamaño reducido, mecanismo simple, buena lubricación.
Diseño Compacto	Tamaño no mayor a 70 x 50 x 30 cm, cantidad mínima de piezas
Calibración Sencilla	Control volumétrico, Raspberry Pi Pico, control de velocidad del motor.
Fácil Uso	Interfaz con el usuario sencilla, mecanismo simple, ingreso de pocos Parámetros (Dos pulsadores de control).
Bajo Costo Operativo	Poco consumo energético, sistemas mecánicos eficientes.
Bajo costo de Mantenimiento	Materiales locales y económicos, disponibles en el mercado local, mecanismo simple.
Consumo Energético	Mecanismo simple, sistema mecánico eficiente
Costo de Producción	Materiales locales, manufactura nacional, pocos elementos, mecanismo simple.
Fácil Ensamble	Pocos elementos, mecanismos simples
Mantenimiento y Repuestos	Manufactura nacional, materiales locales
Medición de Presión y Volumen	Sensores electrónicos, Arduino
Dimensiones	Tamaño no mayor a 70 x 50 x 30 cm, construcción simple, ensamble modular.

Nota. Elaboración propia.

3.1.3 Estudio de la Competencia

Se realizó una investigación sobre los ventiladores mecánicos ya existentes y operativos en el país, que sean producto de diseños independientes, privados o estatales, con el fin de tener en consideración las características con las que ya cuentan dichos dispositivos en el mercado, además que así se tuvo una idea clara del costo de manufactura que podría alcanzar nuestro diseño dependiendo de los componentes que este tuviera, de esta manera se aterrizó el diseño con previo conocimiento del mercado local relacionado a ventiladores mecánicos, se obtuvo información parcial de dos equipos que pertenecen a hospitales que cuentan con UCI, la información se describe en la Tabla 4.

Tabla 4

Especificaciones técnicas generales de ventiladores respiratorios, Perú

Características	Ventilador Respiratorio A	Ventilador Respiratorio B
Marca	Newport	Avante
Modelo	e360	Verdana MC
Precio	10,000 nuevos soles	25,000 nuevos soles
Control	Táctil	Táctil
Monitorización	Interfaz gráfica del usuario	Interfaz gráfica del usuario

Nota. Elaboración propia, mayores detalles en el Anexo A.

3.1.4 Estructura Funcional

Con la ayuda de la estructura funcional del equipo, se puede sacar a flote la serie de procedimientos que va a realizar, primero de manera global, y en sus siguientes niveles se especifica a detalle sus causas y efectos para cada proceso que se vaya a realizar. Es de gran ayuda al momento de diseñar ya que así podemos tener una premisa de los elementos que se utilizará como materiales para construcción del equipo. Como función

principal de la máquina es la suministración de aire, tenemos el esquema de la función global del ventilador mecánico en la Figura 19:

Figura 19

Función global del ventilador mecánico



Nota. Elaboración propia.

En donde como elementos de entrada tenemos:

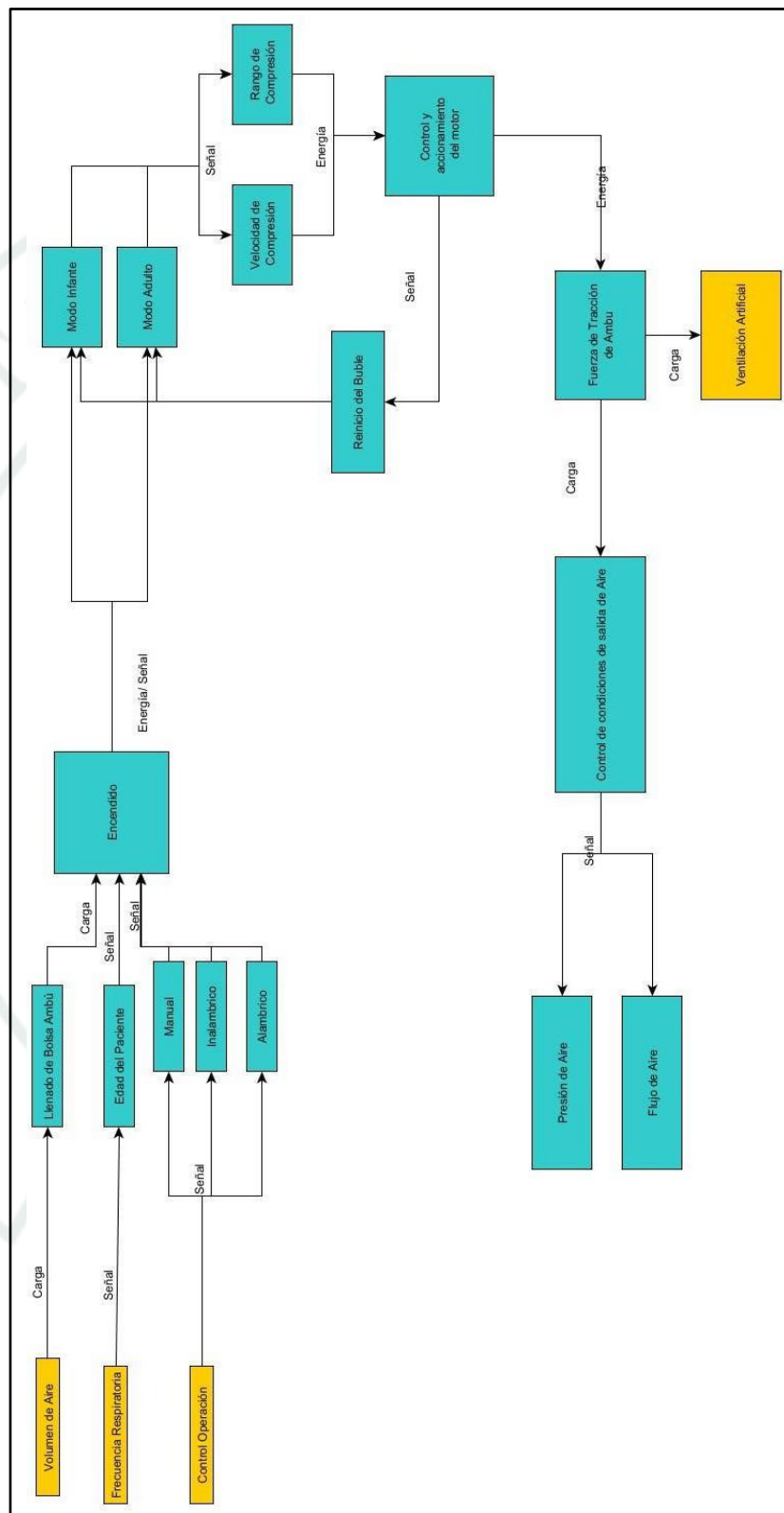
- Materia prima que viene a ser aire.
- Energía eléctrica para el funcionamiento del motor proporcionando de esta manera energía mecánica a la máquina.
- Señal de control, es el encargado de dar la orden de funcionamiento a todo el sistema.

Entre los elementos de salida tenemos:

- Ventilación artificial, en el análisis funcional general se especifican las variables de forma general, más adelante se detalla de manera específica.

En la Figura 20, podemos observar que previo al funcionamiento del motor se va a regular en dos modos de operación, por ende, el volumen y presión del aire dependerán de la velocidad y rango de compresión que reciba el motor de paso a paso.

Figura 20
Análisis de funcionamiento de ventilador mecánico



Nota. Elaboración propia.

3.1.5 Conceptualización de las Posibles Soluciones

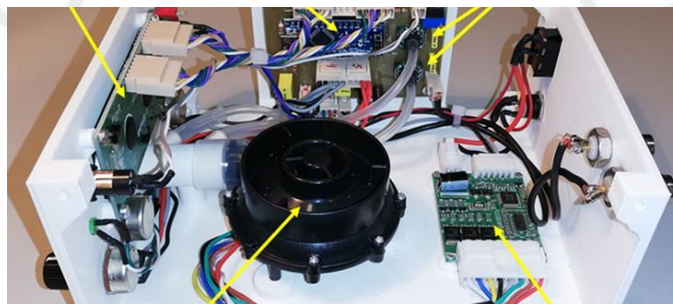
Teniendo claro cómo se podía satisfacer cada requerimiento en función de los objetivos de diseño, se propuso las diferentes soluciones con respecto a los antecedentes:

- a) Dispositivo ventilador con turbina centrífuga
- b) Dispositivo ventilador con leva
- c) Dispositivo ventilador con tornillo sin fin
- d) Dispositivo ventilador mecanismo de engranajes

A continuación, se describe cada dispositivo.

El dispositivo de **ventilador con turbina centrífuga** (Figura 21) funciona gracias al elemento rotatorio soplador que crea una diferencia de presión obligando al flujo de gas a moverse de un punto a otro hacia el paciente, este cuenta con electroválvulas y sensores conectados al sistema de control del dispositivo que permitirá monitorear y definir los parámetros necesarios para llevar a cabo el soporte vital. Investigadores del CIBERES en la Unidad de Biofísica y Bioingeniería de la Universidad de Barcelona han creado un respirador no invasivo de bajo coste y de fácil construcción, destinado a dar soporte a pacientes con insuficiencia respiratoria en zonas de pocos recursos (Ciberes.org, 2020).

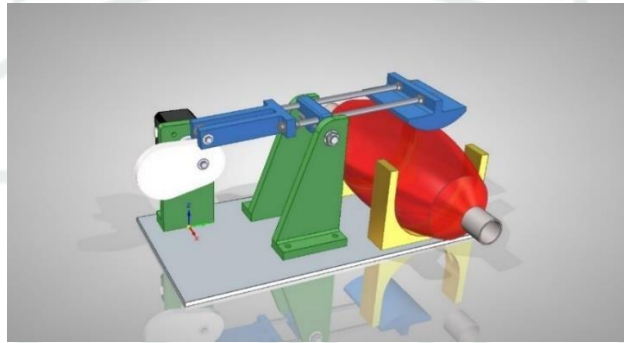
Figura 21
Ventilador con turbina centrífuga



Nota. Ciberes.org, (2020).

El **dispositivo de ventilador con leva**, representado en la Figura 22, funciona gracias al movimiento rotatorio que se le transmite a una leva. Esta leva tiene una forma excéntrica, lo que genera un movimiento oscilatorio en un seguidor unido a una placa. Esta placa es la encargada de presionar al resucitador AMBU, generando un gradiente de presión y, por consecuencia, un movimiento de flujo de aire constante hacia el paciente.

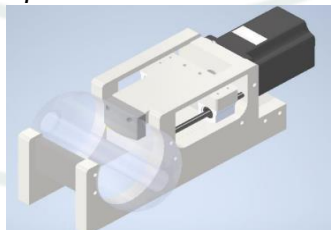
Figura 22
Ventilador con leva



Nota. Elaboración propia.

El **dispositivo ventilador con tornillo sin fin** (Figura 23), consta de dos placas que al juntarse presionan el resucitador AMBU generando el gradiente de presión que obliga el movimiento del flujo de gas, una placa es fija mientras que la otra es móvil, fijada a un tornillo sin fin que rota gracias a un motor eléctrico controlado de forma electrónica.

Figura 23
Ventilador con tornillo sin fin



Nota. Carpio y Sughey, (2022).

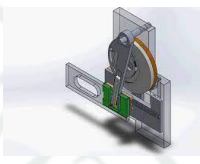
El dispositivo con **Mecanismo de engranajes**, sus dimensiones exteriores son pequeñas, y es eficaz para transmitir esfuerzos con una gran duración y fiabilidad de funcionamiento ya que presenta una relación de transmisión fija. Además, es de fácil mantenimiento, lo inconveniente de este dispositivo es la presencia de ruido a grandes velocidades y el rápido desgaste de la superficie de los dientes.

De la Tabla 5 es posible la identificación de las principales funciones que debe tener la máquina para cumplir con el objetivo de entregar aire, teniendo así:

- Accionamiento del ventilador.
- Compresión.
- Fuente de aire
- Controlador.
- Sensor

Y de las posibles alternativas de solución de uso del mecanismo de ventilación, es factible realizar una matriz morfológica para concretar la solución más eficiente y eficaz, agregando algunos componentes que conforman este sistema de ventilación artificial.

Tabla 5
Matriz morfológica

FUNCIÓN	COMPONENTES
Accionamiento del ventilador.	
Compresión	  
Fuente de aire	 
Controlador.	  
Sensor	  

Nota. Elaboración propia.

3.1.6 Criterios para la Selección de Soluciones

Para elegir la solución más viable del presente proyecto se utilizó la herramienta método ordinal corregido de criterios ponderados, generalmente en la etapa de diseño conceptual, se presentan diferentes alternativas de solución, las cuales deben ser evaluadas con base en criterios establecidos en la cual se determina el orden de preferencia de cada solución desde un punto de vista general , sin tener datos exactos de cada parámetro o propiedad de cada solución (Ruiz et al., 2018), basta con definir los criterios de selección que deben cumplir todas las alternativas, y comparar cada criterio

con los demás para ordenarlos de acuerdo con su relevancia, posteriormente se compara cada alternativa con las demás, en relación con cada criterio, para la comparación se realiza un cuadro donde los criterios o alternativas están ordenadas de la misma forma, tanto en filas como en columnas, siempre se compara el criterio o alternativa de la fila con cada criterio o alternativa de las columnas. Está basado en Tablas donde cada solución se confronta con las demás soluciones asignándose los siguientes valores:

- 1.- Si la solución de la fila es superior a la solución de la columna.
- 0,5.- Si la solución de la fila es igual a la solución de la columna.
- 0.- Si la solución de la fila es inferior a la solución de la columna.

3.1.6.1 Diseño compacto

Este criterio engloba el tamaño del dispositivo, peso, materiales de construcción, la cantidad de piezas o elementos usados, que sea portátil y desmontable, ya que el artefacto debe ser transportado de manera fácil y rápida, además de no ocupar mucho espacio en el área donde será utilizado. Por ende, primeramente, se realiza una inspección de la importancia de cada solución, como se muestra en la Tabla 6.

Tabla 6

Importancia en escala de 0 a 5 de cada solución

Solución	Importancia
A	2
B	4
C	5
D	3

Nota. Elaboración propia.

Posteriormente se compara los valores de la importancia entre las soluciones, como se muestra en la Tabla 7.

Tabla 7*Ponderación de preferencia de cada solución con relación al diseño compacto*

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		0	0	0	1	0.10
B	1		0	1	3	0.3
C	1	1		1	4	0.4
D	1	0	0		2	0.20
TOTAL					10	1

Nota. Elaboración propia.**3.1.6.2 Funcionamiento sencillo**

Principio que determina el uso fácil del artefacto, calibración, mantenimiento y monitoreo, ya que mientras menos operaciones se necesiten para cumplir su función y sea más fácil hacer un seguimiento al paciente y a los elementos que conforman el dispositivo, menos riesgo de fallo deberá tener, lo cual reducirá la frecuencia de mantenimiento preventivo o correctivo (Tabla 8).

Tabla 8*Ponderación de preferencia de cada solución con relación al funcionamiento sencillo*

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		1	1	1	4	0.36
B	0		1	1	3	0.27
C	0	1		1	3	0.27
D	0	0	0		1	0.09
TOTAL					11	1

Nota. Elaboración propia.**3.1.6.3 Alta durabilidad**

Principio que también es un requisito ya antes mencionado, ya que se establece según los materiales de construcción que sean necesarios para la producción local,

presentado una buena calidad y un tiempo de vida útil considerable, también se espera que tenga un recubrimiento bueno para el ambiente donde se ubicará, al mismo tiempo esto optimizará el empleo prolongado del equipo (Tabla 9).

Tabla 9

Ponderación de preferencia de cada solución con relación a alta durabilidad

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		1	0	1	3	0.29
B	0		0.5	1	2.5	0.24
C	0	1		1	3	0.29
D	1	0	0		2	0.19
TOTAL					10.5	1

Nota. Elaboración propia.

3.1.6.4 Costo de operación

Este principio se sujetará de acuerdo a la cantidad de implementos empleados para llevar a cabo la función del dispositivo, el tipo de método que repercute en la energía empleada y por ende en el costo por el uso del dispositivo (Tabla 10).

Tabla 10

Ponderación de preferencia de cada solución con relación al costo de operación.

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		0	0	0	1	0.09
B	1		0	1	3	0.26
C	1	1		1	4	0.35
D	1	1	0.5		3.5	0.30
TOTAL					11.5	1

Nota. Elaboración propia.

3.1.6.5 Costo de manufactura

Este principio se describe tanto con los materiales usados, elementos, tipo de mecanismo, diferentes modos de ventilación, sensores para el monitoreo y sistemas de alerta, ya que mientras más formas tenga y se requiera más elementos para lograrlo, el costo por unidad se acrecentará (Tabla 11).

Tabla 11

Ponderación de preferencia de cada solución con relación costo de manufactura

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		0	0	0.5	1.5	0.15
B	1		0.5	1	3.5	0.35
C	1	0.5		1	3.5	0.35
D	0.5	0	0		1.5	0.15
TOTAL					10	1

Nota. Elaboración propia.

3.1.6.6 Seguridad

Este criterio abarca la seguridad del paciente como del operador del dispositivo, esto se puede reflejar en el sistema de sensores y alarmas que se pueden implementar en el dispositivo (Tabla 12).

Tabla 12

Ponderación de preferencia de cada solución con relación a la seguridad del paciente

Solución	Solución A	Solución B	Solución C	Solución D	S+1	Ponderación
A		0	0	0	1	0.10
B	1		0	0	2	0.20
C	1	1		1	4	0.40
D	1	1	0		3	0.30
TOTAL					10	1

Nota. Elaboración propia.

3.1.7 Conclusiones y Selección de Alternativa

Se asignaron siglas para cada criterio, las cuales son:

- DC: Diseño Compacto.
- FS: Funcionamiento Sencillo.
- AD: Alta Durabilidad.
- CO: Costo de Operación.
- CM: Costo de Manufactura.
- S: Seguridad.

Y en base a las evaluaciones de todas las alternativas disponibles con respecto a sus criterios de evaluación se obtiene la Tabla 13.

Tabla 13

Conclusiones y selección de la mejor alternativa

Solución	DC	FS	AD	CO	CM	S	SUMA	POND
Solución A	0.1	0.36	0.29	0.09	0.15	0.1	1.09	0.18
Solución B	0.3	0.27	0.24	0.26	0.35	0.2	1.62	0.27
Solución C	0.4	0.27	0.29	0.35	0.35	0.4	2.06	0.34
Solución D	0.2	0.09	0.19	0.3	0.15	0.3	1.23	0.21
TOTAL							6	1.00

Nota. Elaboración propia.

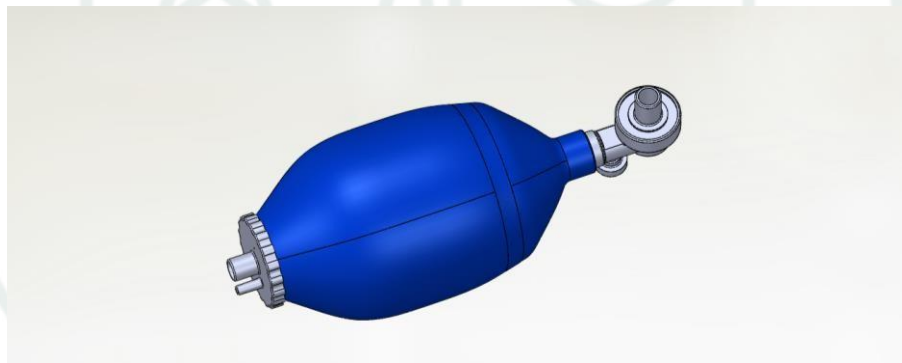
Podemos concluir que la mejor alternativa se presenta en la solución C, donde plantea el desarrollo del prototipo ventilador mecánico con accionamiento de mecanismo sin fin que es particular de los CNC's, con este sistema se tendrá un diseño compacto, un funcionamiento sencillo accionado por un motor de paso, y por otra parte el costo de operación y de la manufactura resulta ser más accesible con respecto a las otras soluciones. Se puede decir que con esta alternativa de solución se tendrá el control del rango y velocidad de contracción de la bolsa de Ambu.

3.1.8 Criterios de la Estructura Impulsor de Aire

La estructura del mecanismo impulsor de aire se diseñará considerando tanto el tamaño como la fuerza de presión necesaria para la expulsión del aire. La bolsa Ambu, gracias a su elasticidad, genera un "efecto globo". Además, en su interior cuenta con pequeños agujeros que garantizan que los usuarios no apliquen una presión de ventilación superior a 45 cm de H₂O (Firma Ambu S.L, 2015).

Antes de proceder con el diseño, se tomaron las medidas de la bolsa de Ambu, como se detalla en el Anexo B. Con estas dimensiones, se llevó a cabo el diseño virtual en el software SolidWorks, como se ilustra en la Figura 24. Para obtener información más detallada, se puede consultar el Anexo C.

Figura 24
Diseño de la bolsa Ambu en SolidWorks

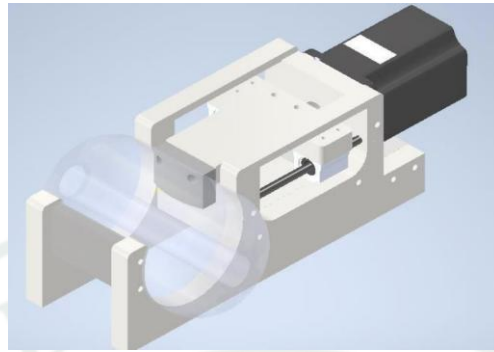


Nota. Elaboración propia.

De entre los diferentes modos de ventilación se seleccionó la solución del dispositivo ventilador con tornillo sin fin, que corresponde a la solución C. El prototipo se tomó de referencia a la Tesis de Sancá y Rodríguez, (2022) donde también se especifica el mecanismo de compresión de la bolsa Ambu con la ayuda del mecanismo del tornillo sin fin, como se muestra en la siguiente Figura 25.

Figura 25

Ventilador artificial con tornillo sin fin



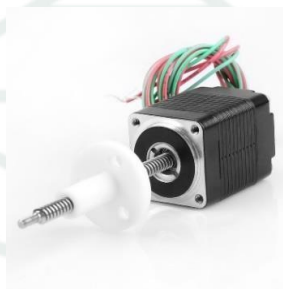
Nota. Carpio y Sughey, (2022).

3.2 Dimensionamiento y Diseño del Tornillo sin fin

Tomando en cuenta la estructura y la tesis de referencia, se puede realizar el control de la velocidad y la posición, por ello el material debe de ser resistente. Además, este tornillo sin fin es muy usado en las impresoras 3D y CNCs, con esa referencia se hará el uso de un motor de pasos, debido a que se necesita velocidades relativamente altas, con poca consideración en el torque (Figura 26).

Figura 26

Tornillo sin fin incrustado a un servomotor



Nota. es.made-in-china.com

3.2.1 Dimensionamiento del Tornillo sin fin

El peso (mg) del tornillo sin fin es de 3N, longitud de 13.7 cm y la carga (P) para considerarse será de 45 N, debido a la resistencia del AMBU a ser comprimido. Este valor fue un estimado de relación la presión máxima en los pulmones del paciente, lo cual se

da en la inspiración y tiene un valor de 45cmH₂O, con un promedio de área de contacto de AMBU. De acuerdo con la existencia en el mercado nacional, se decidió trabajar con un tornillo de potencia de 8 mm de diámetro, ya que este es uno de los más comunes en aplicaciones CNC, cuyas características se muestran a continuación en la Tabla 14:

Tabla 14

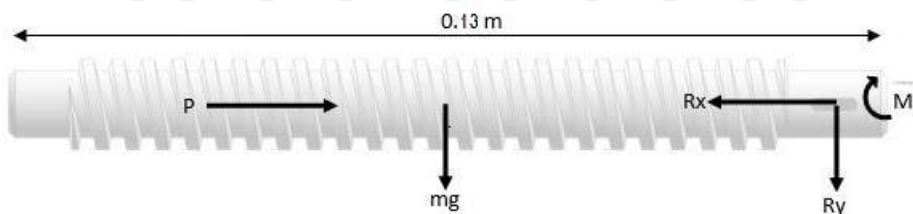
Características del tornillo guía sin fin

Tornillo guía sin fin	
Paso	2 [mm]
Avance	8 [mm]
Longitud	137 [mm]
Material	Acero Inoxidable
Esfuerzo a la fluencia	241 MPa

Nota. Elaboración propia.

Figura 27

Diagrama de cuerpo libre (DCL) del tornillo sin fin



Nota. Elaboración propia.

Para encontrar las reacciones que interactúan en el tornillo sin fin, se realizó el DCL con las fuerzas considerables (Figura 27), el peso del mismo tornillo y la fuerza que comprime el AMBU. Y mediante la primera Ley de Newton se determinaron las ecuaciones:

$$R_y = mg \quad (4)$$

$$R_x = P \quad (5)$$

$$M = mg * 0.1 \quad (6)$$

El esfuerzo axial, debido a la forma de eje que posee y la aplicación de la carga en esa dirección. El pandeo no se considera debido a la baja esbeltez, baja carga aplicada y el material resistente.

3.2.1.1 Esfuerzo a la tensión

Para ser conservadores al determinar el esfuerzo de tensión se consideró un área de esfuerzo por tensión (A_f) como el círculo formado con el diámetro interno (d_r) del tornillo preseleccionado, por tanto, el esfuerzo es el siguiente:

$$\sigma_t = \frac{F}{\frac{\pi}{4} d_r^2} \quad (7)$$

Donde:

- F es la fuerza necesaria para poder presionar el AMBU, este valor se estima en unos 45 N.
- d_r es el diámetro interno del tornillo correspondiente a 6 mm ya que al diámetro externo se le resta 2 veces la profundidad de la rosca que en este caso es 1mm.

Reemplazando datos en la ecuación:

$$\sigma_t = \frac{45N}{\frac{\pi}{4} (6 - 2)^2} = 1,76 \text{ MPa}$$

3.2.1.2 Esfuerzo cortante

El esfuerzo cortante tiene relación con el barrido de la tuerca a través de las cuerdas del tornillo de potencia y no sólo se analiza en el tornillo, sino también en la

tuerca. De forma que tenemos las siguientes ecuaciones para el tornillo y para la tuerca en ese orden:

$$\tau_s = \frac{F}{0.5\pi p d} \quad (8)$$

Donde:

- d corresponde al diámetro exterior de la tuerca con 8 mm
- p corresponde al paso, es decir 2 mm

Reemplazando en la ecuación:

$$\tau_s = \frac{45N}{0.5\pi(2 \times 10^{-3})(8 \times 10^{-3})} = 1,79 \text{ MPa}$$

3.2.1.3 Esfuerzo a la torsión

Sólo se considera el esfuerzo de torsión cuando existe una fricción relevante entre la tuerca y el tornillo, en caso de no existir lubricación o estén oxidados. La ecuación que describe este fenómeno es la siguiente:

$$\tau = \frac{16T}{\pi d_r^3} \quad (9)$$

Reemplazando variables para el esfuerzo a la torsión:

$$\tau = \frac{16(0,072 \text{ Nm})}{\pi(6 \times 10^{-3})^3} = 1,69 \text{ MPa}$$

Donde τ corresponde al torque aplicado en el tornillo guía sin fin para hacerlo girar, cuyo valor será determinado en la sección 3.3.1.4 a continuación.

3.2.1.4 Esfuerzo flexionante en la raíz de la rosca

El esfuerzo flexionante en la raíz de la rosca se calcula como:

$$\sigma_b = \frac{6F}{\pi d_r n_t p} \quad (10)$$

Para un correcto análisis se emplea la teoría de energía de distorsión, calculando el esfuerzo de Von Mises, el cual consta de componentes de esfuerzo xyz tridimensionales, se debe tomar en consideración que $\sigma_x = \sigma_b$, $\sigma_z = \sigma_t$, $\tau_{yz} = \tau$ y $\sigma_y = \tau_{xy} = \tau_{zx} = 0$.

$$\sigma_b = \frac{-6F}{\pi * d_r * 0.063}$$

$$\sigma_b = \frac{-6 * 45N}{\pi * 6 * 10^{-3}mm * 0.063} = 0,227 \text{ Mpa}$$

$$\sigma_y = 0 \quad \tau_{yz} = 1,69 \text{ MPa}$$

$$\sigma_z = 1,77 \text{ Mpa} \quad \tau_{zx} = 0$$

Posteriormente se realiza el cálculo, para ello se usa la siguiente ecuación.

$$\sigma = \frac{1}{\sqrt{2}} * [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]^{1/2}$$

Reemplazando datos en la ecuación:

$$\sigma = \frac{1}{\sqrt{2}} * [(0,227 \text{ Mpa} - 0)^2 + (0 - 1,77 \text{ Mpa})^2 + (1,7 \text{ Mpa} - 0,227 \text{ MPa})^2 + 6(1,69 \text{ Mpa})^2 + 0]^{1/2}$$

$$\sigma = 2,37 \text{ MPa}$$

Finalmente, al conocer todos los valores se deben comparar los mismos con la propiedad mecánica del material y obtener un factor de seguridad, el cual nos asegure que el tornillo de potencia cumple los requerimientos.

$$n = \frac{s}{y_{\sigma}} \quad (11)$$

Reemplazando variables en la ecuación:

$$n = \frac{241 \text{ Mpa}}{2,37 \text{ Mpa}} = 101,63$$

Este factor de seguridad está muy por encima de cualquiera considerado en diferentes aplicaciones de ingeniería o recomendado por ASME, sin embargo, es el más comercial y diseñar uno menos sobredimensionado implicaría mayores costos.

3.3 Dimensionamiento del Motor

En el punto anterior se calculó el torque necesario para hacer girar el tornillo de guía sin fin, mismo dato que se necesita para seleccionar el motor a ser usado, además de las revoluciones por minuto de este.

La velocidad es el factor es de mucha importancia, ya que con la velocidad podemos tener las dos opciones de operación y dar solución a nuestro objetivo de esta investigación por ende la ecuación se da:

$$RPM = \frac{l}{t * L * 60} \quad (12)$$

Donde:

- l es la longitud para comprimir el AMBU, en este caso tomamos la mayor longitud a ser comprimido, definido por los diseñadores en 70 mm.

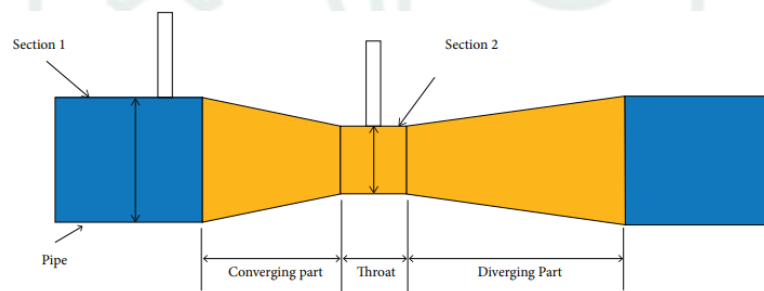
- L es el avance del tornillo de potencia por revolución, 8mm/rev
- t es el tiempo de inspiración o espiración, para este cálculo se toma el menor tiempo ya que eso implica la mayor velocidad, es decir 2 s.

$$RPM = \frac{70 \text{ mm}}{2 \text{ seg} * \frac{8 \text{ mm}}{\text{rev}}} * \frac{60 \text{ seg}}{\text{min}} = 263 \frac{\text{rev}}{\text{min}}$$

3.4 Dimensionamiento del Método de Tubo de Venturi

Antes de realizar los cálculos, cabe mencionar que se tomó de referencia el artículo (Kshetry et al., 2021), donde ellos usaron para medir la presión y el flujo de aire que se emana al accionar la bolsa de Ambu (Figura 28).

Figura 28
Tubo de Venturi

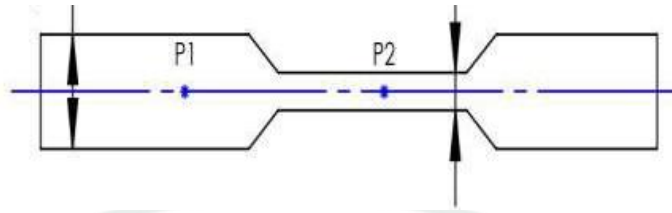


Nota. Kshetry et al., (2021).

3.4.1 Dimensionamiento

La ecuación de Bernoulli relaciona la presión y la velocidad de dos puntos denominados (P_1 y P_2) en un flujo laminar de aire constante de densidad 1 (Khan Academy, 2022). Los diámetros son de 70 mm y 208 mm respectivamente con una longitud de 14 cm, por ende, escribimos la ecuación de Bernoulli de la siguiente manera:

Figura 29
Tubo de Bernoulli



Nota. Tomado de Khan Academy, (2022).

$$P_1 + \frac{1}{2} \rho V_1^2 + \rho g z_1 = P_2 + \frac{1}{2} \rho V_2^2 + \rho g z_2 \quad (13)$$

$$z_1 = z_2$$

$$P_1 + \frac{1}{2} \rho V_1^2 = P_2 + \frac{1}{2} \rho V_2^2$$

$$P_1 - P_2 = \frac{1}{2} \rho V_2^2 - \frac{1}{2} \rho V_1^2$$

$$P_1 - P_2 = \frac{1}{2} \rho V_2^2 - \frac{1}{2} \rho V_1^2$$

$$Q_1 = V_1 * A_1$$

$$Q_2 = V_2 * A_2$$

$$Q_1 = Q_2$$

$$V_1 * A_1 = V_2 * A_2$$

$$A_1 = \frac{\pi * \phi_1^2}{4}$$

$$A_2 = \frac{\pi * \phi_2^2}{4}$$

$$V_1 * \frac{\pi * \phi_1^2}{4} = V_2 * \frac{\pi * \phi_2^2}{4}$$

$$V_1 * \phi_1^2 = V_2 * \phi_2^2$$

Los diámetros del tubo interior se encuentran en los planos de diseño del Anexo

C3: Tubo de Venturi, reemplazando valores obtenemos lo siguiente:

$$\phi_1 = 20,80 \text{ mm} \quad \phi_2 = 7 \text{ mm}$$

Por lo tanto: $V_2 = 8.83V_1$

$$\Delta P = P_1 - P_2 = \frac{\rho * ((8.83V_1)^2 - V_1^2)}{2} \quad (14)$$

$$\Delta P = \frac{\rho * (76.96V_1^2)}{2}$$

$$V_1 = \sqrt{\frac{2 * \Delta P}{\rho * 76.96}}$$

Una vez que se calculó la velocidad del aire, en uno de sus puntos, a partir de ello se puede obtener el caudal (Subdirección General de Administración del Agua, 2001)

$$Q = A_1 * V_1 = \text{constante} \quad (15)$$

En donde:

- A: es el área hidráulica del conducto (m²)
- V: es la velocidad en la tubería (m/s)

Como además se tiene que: $Q = \text{volumen} / \text{tiempo}$

- *tiempo = periodo de señal*
- *volumen = $Q * \text{periodo de señal}$ [unidades de volumen]*

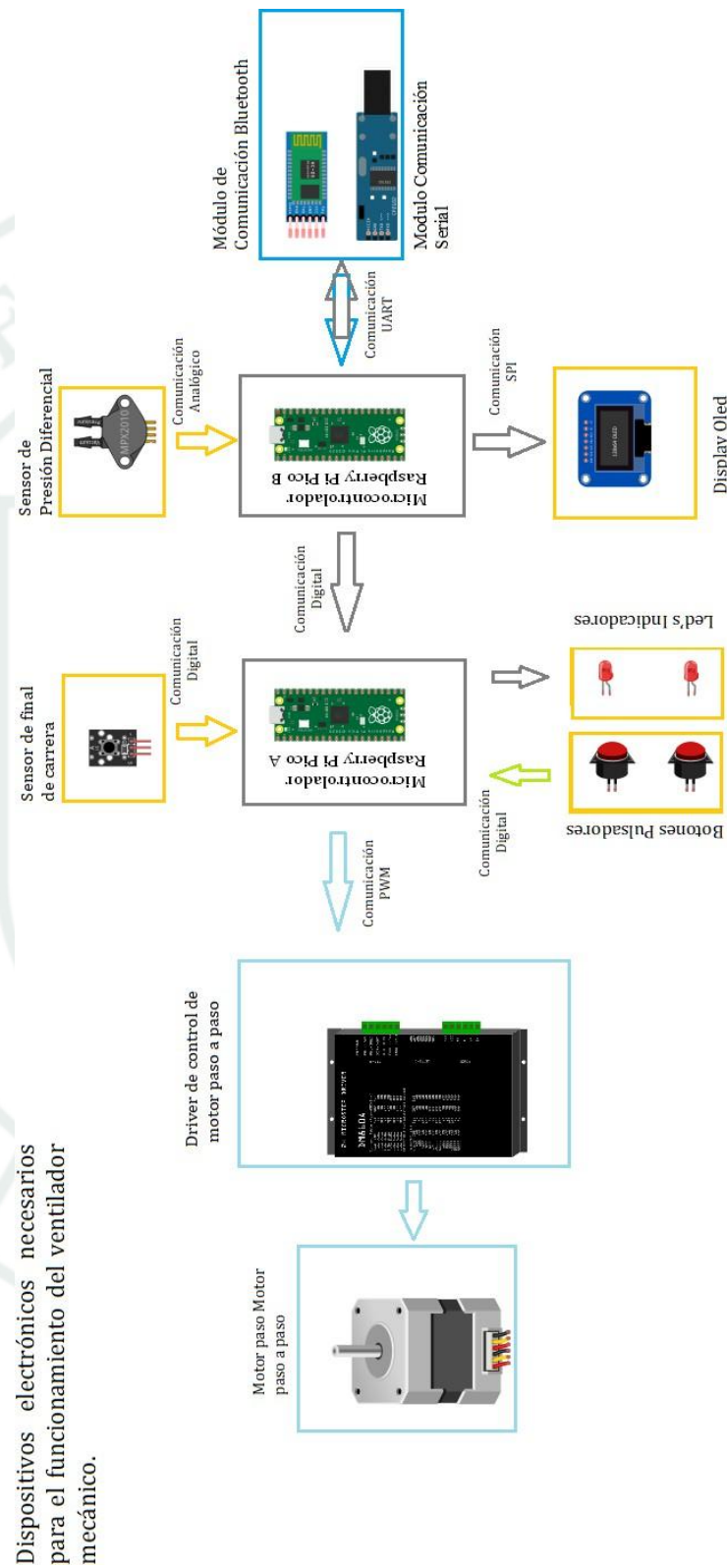
3.5 Instrumentación del Ventilador Mecánico

En este apartado se invoca a los dispositivos electrónicos necesarios para poner en marcha el prototipo, a partir de la solución C. Por lo tanto, la instrumentación comprende a todos los dispositivos electrónicos necesarios para el funcionamiento del prototipo de acuerdo al objetivo planteado en el primer capítulo, como son los sensores, actuadores, microcontroladores y otros.

En la Figura 30, se muestra algunos de los dispositivos electrónicos del ventilador mecánico, también se menciona a la comunicación de datos entre estos, estos diálogos son de forma direccional y bidireccional por el protocolo de comunicación que estos presentan. Cabe acotar que el ventilador mecánico tendrá 2 modos de operación, es por ello que en el esquema se muestra: el módulo de comunicación serial UART, módulo de comunicación Bluetooth y los botones para la operación manual.

Figura 30

Esquema de dispositivos electrónicos del ventilador mecánico



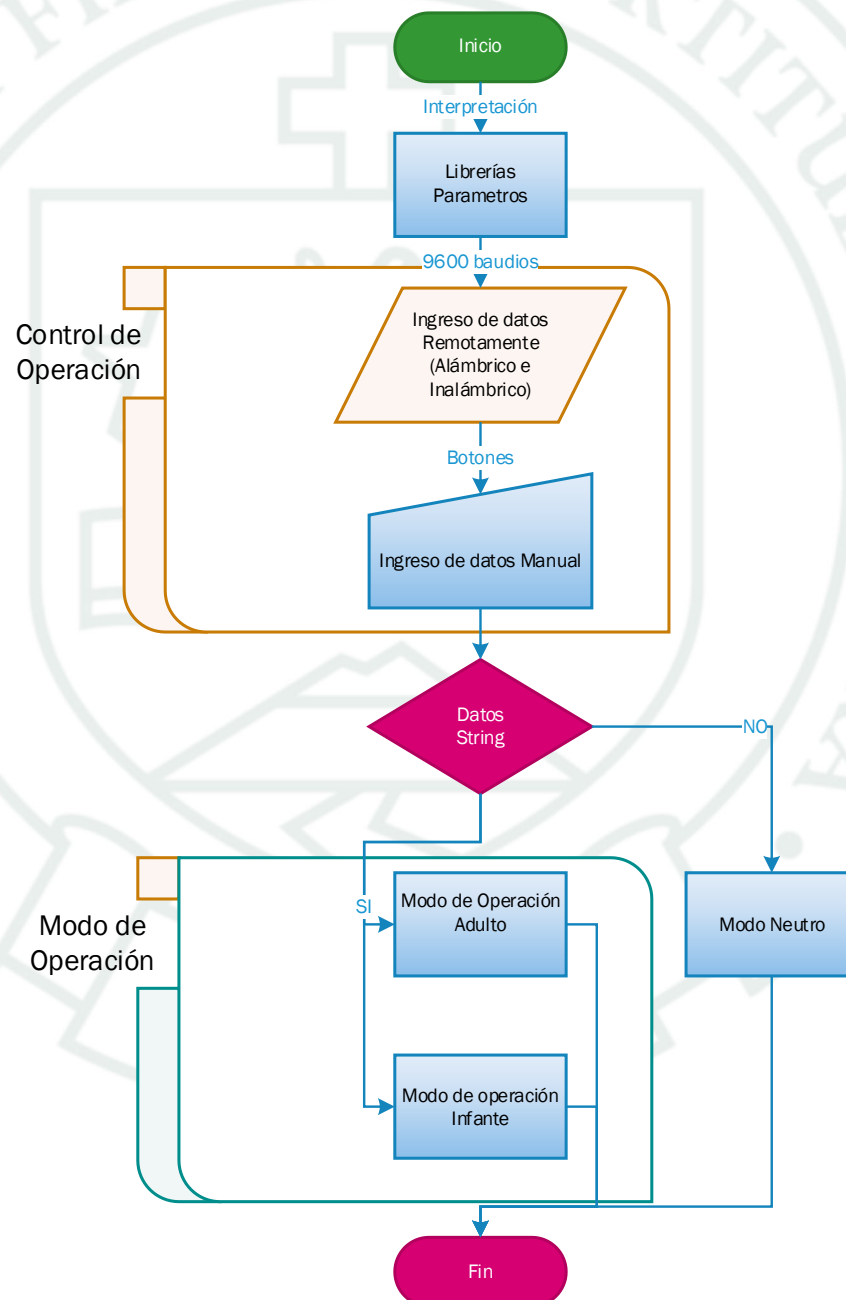
Nota. Elaboración propia.

3.6 Sistema de Control del Ventilador

El ventilador mecánico posee dos modos de operación, el modo infante y el modo adulto, por ende, estos modos de operación deben de ser controlados por dos modos, manual y remoto, el modo remoto compromete la comunicación serial alámbrica e inalámbrica (Figura 31).

Figura 31

Diagrama de flujo de control y operación del ventilador



Nota. Elaboración propia.

3.7 Adquisición de Dispositivos Biomédicos y Electrónicos

3.7.1 Bolsa de Ambu

Es un dispositivo manual para proporcionar ventilación con presión positiva para aquellos pacientes que no respiran o que no lo hacen de forma adecuada. El dispositivo con lo que se trabaja está estandarizado en la norma ISO-10651-4. El dispositivo médico tiene 1650 ml de volumen y de 800 ml de expulsión de volumen de aire; por otra parte, la presión de aire que genera es de 60 cmH₂O, las dimensiones de la bolsa de Ambu que se muestra en la Figura 32 es de 325 x 132mm.

Desde el punto de vista técnico, está compuesta por una bolsa autoinflable, una válvula unidireccional que dirige el flujo de aire hacia el paciente, una válvula de alivio de presión que limita la presión máxima suministrada, una mascarilla facial o conexión a tubo endotraqueal, un reservorio de oxígeno y una entrada para suministro de O₂. Algunos modelos incluyen además un puerto para válvula PEEP y conexiones para monitoreo.

En cuanto a sus características físicas, la bolsa suele estar fabricada en silicona médica o materiales poliméricos como PVC o SEBS, los cuales pueden ser reutilizables o desechables, y presentan propiedades de elasticidad, resistencia y biocompatibilidad. Su diseño es generalmente semitransparente, lo que permite la inspección visual del funcionamiento durante su uso.

Técnicamente, una bolsa AMBU para adulto presenta un volumen aproximado de entre 1600 y 2000 ml, con un volumen tidal entregado que oscila entre 500 y 800 ml por compresión. La válvula de alivio de presión está calibrada típicamente en un rango de 30 a 45 cmH₂O, lo que garantiza la protección de los pulmones frente a sobrepresiones. Asimismo, puede alcanzar concentraciones de oxígeno cercanas al 90–100% cuando se utiliza con reservorio y suministro de oxígeno externo.

Desde el punto de vista funcional, el dispositivo incorpora un sistema de válvulas unidireccionales que evita la reinhalación de dióxido de carbono, asegurando una ventilación eficiente. Además, cuenta con conectores estándar de 15 y 22 mm, lo que permite su compatibilidad con diversos dispositivos médicos. Su diseño ergonómico facilita el agarre y reduce la fatiga del operador durante su uso prolongado.

En términos de seguridad, la bolsa AMBU integra mecanismos de protección como la válvula de alivio de presión, materiales libres de látex y sistemas que reducen el riesgo de contaminación cruzada.

La válvula de alivio de la bolsa tipo AMBU presenta una **resistencia mecánica de tipo elástico**, generalmente proporcionada por un resorte o un diafragma flexible. Esta resistencia actúa como un elemento de oposición al flujo de aire, manteniendo la válvula en estado cerrado bajo condiciones normales de operación.

El principio de funcionamiento se basa en el equilibrio de fuerzas: la presión interna del sistema ejerce una fuerza sobre la superficie efectiva de la válvula, la cual es contrarrestada por la fuerza elástica del elemento mecánico. Cuando la presión supera un umbral preestablecido, la fuerza generada por el gas vence la resistencia mecánica, provocando la apertura de la válvula y permitiendo la liberación del exceso de presión.

De esta manera, la resistencia mecánica no solo está presente, sino que es un parámetro fundamental en la calibración del dispositivo, ya que define la presión máxima de operación segura del sistema.

Figura 32
Bolsa de Ambu



Nota. Elaboración propia.

3.7.2 Guantes Látex

Los guantes de látex quirúrgico en este trabajo de investigación tienen la función de reemplazar a los pulmones del paciente. En la Figura 33 se muestra un guante de tamaño mediano con un largo de 240mm y 100 mm de ancho.

Figura 33
Guante quirúrgico talla M



Nota. Elaboración propia.

3.7.3 Regulador de Oxígeno

El regulador de oxígeno YH. YX11A al que se pudo tener acceso, dispone de un regulador de flujo Q de 0 a 10 L/min para presiones de 15MPa de entrada y 0.2 MPa de salida como se muestra en la Figura 34.

Figura 34
Regulador de oxígeno YH. YX11A



Nota. Elaboración propia.

3.7.4 Sensor de Presión Diferencial

El sensor que se muestra en la Figura 35 es el MPX5010 DP, es un sensor de presión diferencial que funciona bajo el principio de variación de resistencia. El sensor opera a 5V de corriente continua, puede leer desde 0kPa hasta los 10kPa y, su tiempo de respuesta es de 1 ms, con una sensibilidad de 450mV/mm según la hoja del fabricante (alldatasheet.com, 2012) y Anexo B2: Sensor MPX5010DP.

Figura 35
Sensor MPX5010DP



Nota. Elaboración propia.

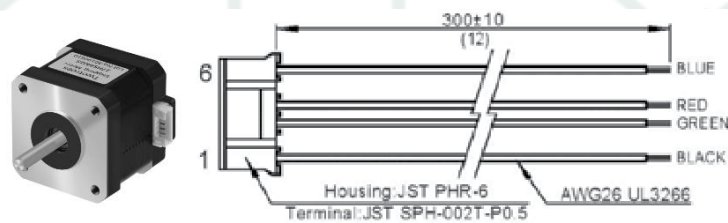
3.7.5 Motor Paso a Paso

El motor paso a paso Nema 17 es de tipo bipolar, con un ángulo de paso de $1,8^\circ$, es decir, puede dividir cada una de las revoluciones o vueltas en 200 pasos. Cada bobinado de los que tiene en su interior soporta 1.2A de intensidad a 4v de tensión, con lo que es capaz de desarrollar una fuerza considerable de 3.2 kg/cm, mayores detalles en el Anexo

B3: Motor nema 17. En el apartado de Dimensionamiento del Motor se estima que el motor se aproxime a los 200 pasos, bajo esas condiciones este dispositivo es ideal para la compresión de la bolsa de Ambu.

La numeración de los pines del motor Nema 17 va en el orden ascendente como se muestra en la Figura 36, esto es muy importante para realizar la conexión con el Driver de control.

Figura 36
Motor Nema 17 paso a paso



Nota. Elaboración propia.

3.7.6 Tornillo Guía sin fin

Este tornillo es propio de las impresoras 3D (Figura 37), en los cálculos aproximados del apartado de Dimensionamiento del Tornillo sin fin, se concluyó usar este por la facilidad de adquisición en el mercado local. Las especificaciones que presenta este son:

- Material: Acero inoxidable
- Diámetro del tornillo de plomo: 8mm
- Paso del tornillo: 2mm
- Plomo de rosca: 2mm mayores detalles en el Anexo B4: Tornillo Sin fin CNC.

Figura 37

Tornillo sin fin (CNC)



Nota. Elaboración propia.

3.7.7 Sensor de Final de Carrera

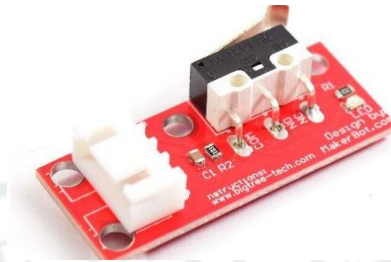
El final de carrera mecánico tiene como principal componente un Switch mecánico tipo bumper, el cual al golpear un objeto cierra el contacto normalmente abierto (Figura 38). Son de fácil uso e instalación comparado con un Final de carrera óptico, poseen una vida útil de aprox. 1 millón de usos. Internamente pueden tener 2 estados, en NC (normalmente cerrados) o NA (normalmente abiertos), lo que quiere decir que en estado de reposo (sin pulsar sobre el final de carrera) el contacto va a estar abierto (NA) o cerrado (NC) (Tostatronic, 2016).

Características del dispositivo:

- Voltaje de alimentación: 5V DC
- Tipo: Limit switch
- Conector 4 cables: 5V, 0V, 0V, SALIDA (ON: 0V, OFF:5V)
- Modelo: Mech Endstop v1.2
- Diseñado para el Shield 3D Ramps 1.4

Figura 38

Sensor final de carrera mecánico



Nota. Tostatronic, (2016).

3.7.8 Driver de Control

Es un controlador completamente digital paso a paso, diseñado con un algoritmo de control DSP avanzado que aprovecha la tecnología más reciente en control de movimiento. Este controlador ha alcanzado un nivel excepcional de suavidad en el sistema, garantizando un par de torsión óptimo y eliminando la inestabilidad en el rango medio.

Figura 39

Driver de control



Nota. Naylamp Mechatronics, <https://naylampmechatronics.com/driver-pap-stepper/558-driver-pap-dm542t-4a.html>

3.7.9 Raspberry Pi Pico

Placa electrónica de microcontrolador de alta performance; contiene un microcontrolador RP2040, dos núcleos ARM Cortex M0+ con velocidades de reloj de

hasta 133 MHz, 264KB de memoria SRAM y 2MB de almacenamiento en memoria Flash, 26 pines GPIO multifunción, Interfaces SPI, 3ADC de 12 bits para conexión a sensores con salidas analógicas, Sensor de temperatura integrado en el microcontrolador, además es programable en C/C++ y MicroPython.

Figura 40
Raspberry Pi Pico



Nota. Creatividad AHORA, <https://creatividadahora.com/producto/raspberry-pi-pico/>

3.7.10 Display Oled

Es una pequeña pantalla OLED de 128*64 pixeles que facilitan el control de cada pixel individualmente, usada para mostrar gráficos reducidos en equipos portátiles; tienen gran contraste, el consumo de energía es mínimo. Trabaja con sistemas de 3.3V a 5V sin necesidad de convertidores. (Naylamp Mechatronics)

Figura 41
Display Oled



Nota. Naylamp Mechatronics,
<https://naylampmechatronics.com/oled/850-display-oled-096-i2c-12864-ssd1306.html>

3.7.11 Módulo Bluetooth

Es una placa configurable como transmisor o receptor, usa un voltaje de entrada de 3.3-6V, la velocidad de transmisión serial de 9600, las dimensiones del módulo del núcleo son: 28mmx 15mm x 2.35mm; utiliza un sistema de navegación GPS y contiene regulador de voltaje 3.3V y 150mA. (Tesla Electronic)

Figura 42
Módulo Bluetooth

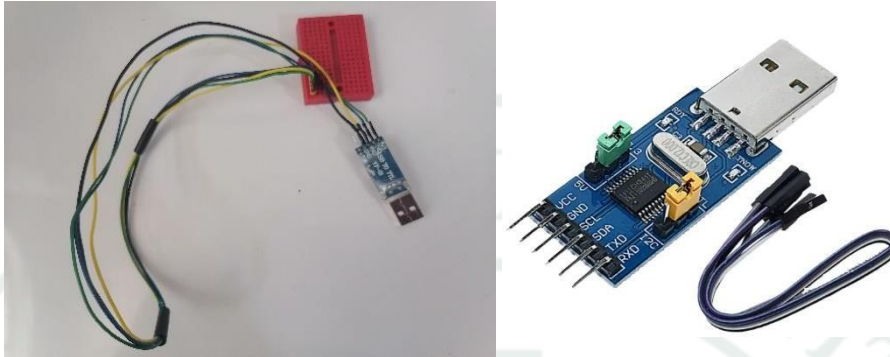


Nota. Tesla Electronic, <https://www.teslaelectronic.com.pe/producto/modulo-bluetooth-hc-05/>

3.7.12 Módulo Serial USB

Es un controlador de serie USB-I2C/SPI/1-Wire; en otras palabras, es un chip de control que permite al USB conectarse con otros dispositivos mediante diferentes protocolos de comunicación; usualmente es usado en aplicaciones que requieren conexión de USB con microcontroladores o placas de control como son Arduino o Raspberry Pi, su uso puede incluir sistema para control de movimiento, medición, automatización o control de posiciones. (Compel S.A)

Figura 43
Módulo serial USB



Nota. COMPEL S.A., <https://www.compelectronica.com/product/modulo-2-en-1-usb-a-i2c-iic-uart-y-usb-a-ttl-ch341t/>

3.7.13 Tubo Flexible de 1/8 y 1 ½

Es un tubo flexible de acero galvanizado revestido con PVC, se usa para la protección de cables eléctricos y fibra óptica que son sometidos a esfuerzos mecánicos, es resistente a rayos ultravioletas, aceites, vapores corrosivos; así mismo la máxima temperatura de operación es de 55°C. (Promelsa)

Figura 44
Tubo flexible de 1/8 y 1 ½



Nota. <https://www.promelsa.com.pe/1029325-tuberia-flexible-liquid-tight-con-forro-de-pvc-1-1-2-x50mts.html>

3.7.14 Fuente de Alimentación Conmutada

Una fuente conmutada es un equipo electrónico que convierte energía eléctrica utilizando transistores en estado de conmutación. A diferencia de un regulador de voltaje que emplea transistores en su región activa de amplificación, las fuentes conmutadas activamente alternan estos transistores a altas frecuencias (generalmente entre 20 y 100 kilohercios) entre los estados de corte (abierto) y saturación (cerrado).

Figura 45
Fuente de alimentación conmutada



Nota. HI-FI, <https://hifisac.com/shop/s-72w-12-fuente-de-alimentacion-conmutada-de-12-vdc-6-a-72-w-transformador-ca-220-v-a-dc-24-v-1855#attr=>

3.8 Cumplimiento de las normas

3.8.1 Norma ISO 10651-4

La ISO 10651-4 es una norma internacional que establece los requisitos para resucitadores manuales, comúnmente conocidos como bolsas AMBU, utilizados para ventilación pulmonar en pacientes.

Esta norma aplica a:

- ✓ Dispositivos de ventilación manual.
- ✓ Uso en pacientes adultos, pediátricos y neonatales.
- ✓ Equipos utilizados en emergencias, transporte y reanimación.

La normal define criterios en los siguientes aspectos:

Rendimiento:

- ✓ Volumen de ventilación adecuado.
- ✓ Capacidad de suministro de aire/oxígeno.
- ✓ Control de presión.

Seguridad:

- ✓ Presencia de válvula de alivio de presión.
- ✓ Prevención de sobrepresión pulmonar.
- ✓ Funcionamiento confiable bajo uso continuo.

Válvulas y flujo:

- ✓ Válvulas unidireccionales eficientes.
- ✓ Flujo adecuado de aire sin fugas significativas.

Materiales:

- ✓ Materiales biocompatibles.
- ✓ Resistentes a desinfección y limpieza.
- ✓ No tóxicos para el paciente.

Tabla 15

Cumplimiento de la norma ISO 10651-4 en el prototipo.

Requisito ISO 10651-4	Descripción de la norma	Implementación en el prototipo	Cumplimiento
Volumen de ventilación	Debe proporcionar volumen adecuado al paciente Adulto: 400–700 ml Niño: 200–400 ml	Control de volumen mediante compresión mecánica. AMBU1600–2000 ml, ajustado por recorrido del mecanismo	Sí
Control de presión	Evitar sobrepresión pulmonar. Adulto: 30–45 cmH₂O Niño: 20–30 cmH₂O	Sensor MPX5010DP monitorea presión. (0–10 kPa $\approx$ 0–102 cmH ₂ O) permite monitoreo en ambos rangos	Sí
Válvula de alivio	Limita la presión máxima: Adulto: 35–45 cmH₂O Niño: 30–35 cmH₂O	Depende de la bolsa AMBU ya que la válvula de alivio se puede regular.	Sí
Flujo de aire	Flujo continuo, sin fugas y monitoreo. Adulto: 6–10 L/min Niño: 4–8 L/min	Generada por compresión controlada y medida por el tubo de Venturi	Sí
Válvulas unidireccionales	Evitan reflujo de aire (fugas permisibles: < 50 ml/min)	Integradas en la bolsa AMBU	Sí
Materiales biocompatibles	Materiales grado médico y no tóxicos (cumplimiento ISO 10993)	AMBU comercial (silicona/PVC)	Sí
Durabilidad	Resistencia al uso continuo. $\geq$ 10,000 ciclos de operación.	Limitado al uso experimental	Sí
Pruebas y validación	Ensayos clínicos y técnicos. Precisión de parámetros $\pm$ 10%	No implementados (prototipo académico)	Parcial

Nota. Elaboración propia.

La selección de la bolsa AMBU se basa en la norma ISO 10651-4, la cual establece requisitos de diseño, seguridad y rendimiento para garantizar una ventilación adecuada. El prototipo cumple parcialmente con esta norma, apoyándose en una bolsa AMBU comercial que incorpora válvulas de seguridad y cumple estándares médicos.

No obstante, al tratarse de un sistema experimental, no cuenta con validación clínica ni certificación formal.

3.8.2 Norma ISO 80601-2-12:2020

Esta norma establece los requisitos para garantizar la seguridad y el funcionamiento esencial de ventiladores mecánicos utilizados en cuidados críticos, esta norma define:

Funcionamiento esencial:

- ✓ Capacidad de suministrar ventilación controlada.
- ✓ Precisión en parámetros como presión, volumen y flujo.
- ✓ Estabilidad en la operación del equipo.

Seguridad del paciente:

- ✓ Protección contra sobrepresión.
- ✓ Sistemas de alarma ante fallas.
- ✓ Prevención de riesgos durante la ventilación.

Control de parámetros:

- ✓ Regulación de frecuencia respiratoria.
- ✓ Control del volumen Tidal.
- ✓ Ajuste de presión inspiratoria y espiratoria.

Sistema de alarma:

- ✓ Alarmas por presión alta/baja.
- ✓ Alarmas por desconexión.
- ✓ Alarmas por fallo del sistema.

Seguridad eléctrica:

- ✓ Protección contra descargas eléctricas.

- ✓ Aislamiento de componentes.
- ✓ Cumplimiento de normas eléctricas médicas.

Compatibilidad electromagnética:

- ✓ Funcionamiento sin interferencias.
- ✓ No afectar otros equipos médicos.

Ensayos y validación:

- ✓ Pruebas de desempeño.
- ✓ Ensayos de precisión.
- ✓ Validación del funcionamiento continuo.

Usabilidad:

- ✓ Interfaz clara para el usuario.
- ✓ Facilidad de operación.
- ✓ Minimización de errores humanos.

Con el fin de evaluar el desempeño del sistema, se realizó una comparación con los requisitos establecidos en la norma ISO 80601-2-12. Se determinó que el prototipo cumple parcialmente con los criterios de funcionamiento esencial, control de parámetros y usabilidad, debido a la implementación de un sistema de control basado en microcontroladores y sensores.

No obstante, al tratarse de un prototipo experimental, no se han implementado sistemas avanzados como alarmas clínicas, validación mediante ensayos estandarizados ni evaluación de compatibilidad electromagnética, por lo que su cumplimiento con la norma es de carácter referencial.

Tabla 16*Cumplimiento de la norma ISO 80601-2-12:2020 en el prototipo.*

Requisito de la norma	Descripción	Implementación en el prototipo	Cumplimiento
Funcionamiento esencial	Ventilación continua y controlada	Control mediante microcontrolador + motor paso a paso	Sí
Control de parámetros	Ajuste de presión, volumen y frecuencia	Control presión mediante sensor MPX5010DP y medición de volumen mediante tubo de Venturi	Sí
Volumen tidal	Precisión en volumen suministrado	Control indirecto mediante compresión de AMBU	Sí
Sistemas de alarma	Alarmas por fallas o valores críticos	No implementado	No
Seguridad del paciente	Protección contra riesgos respiratorios	Limitado (uso de AMBU y botón de seguridad)	Parcial
Seguridad eléctrica	Protección contra descargas	Regulación de voltaje y fuentes separadas	Parcial
Compatibilidad electromagnética	No interferir con otros equipos	No evaluado	No
Usabilidad	Interfaz clara y fácil de usar	Pantalla OLED + control remoto	Sí
Validación y ensayos	Pruebas clínicas y de laboratorio	No realizado	No

Nota. Elaboración propia.

3.8.3 NTP IEC 60601-1

La norma NTP IEC 60601-1 establece los requisitos generales de seguridad básica y funcionamiento esencial de equipos médicos eléctricos, con el objetivo de proteger al paciente, al operado y al entorno. Esta norma evalúa los siguientes puntos:

- ✓ Seguridad eléctrica: Protección contra descargas eléctricas.
- ✓ Aislamiento: Separación entre partes energizadas y el paciente.
- ✓ Protección contra sobre corriente y sobrecalentamiento.
- ✓ Diseño seguro: Reducción de riesgos mecánicos y eléctricos.
- ✓ Funcionamiento esencial: El equipo debe operar sin fallas críticas.
- ✓ Protección en condiciones anormales: El sistema debe ser seguro ante fallos.

Se realizó una evaluación del prototipo en base a los lineamientos establecidos en la norma NTP IEC 60601-1, identificándose un cumplimiento parcial en aspectos de seguridad eléctrica y funcionamiento esencial. Si bien el sistema incorpora elementos de protección básica y opera con niveles de tensión reducidos, no se han implementado ensayos normalizados ni mecanismos completos de protección ante fallas, debido a su naturaleza experimental.

Tabla 17

Cumplimiento de la norma NTP IEC 60601-1 en el prototipo.

Requisito de la norma	Descripción	Implementación en el prototipo	Cumplimiento
Seguridad eléctrica	Protección contra descargas eléctricas	Uso de fuentes DC (12V y 24V) y reguladores	Sí
Aislamiento eléctrico	Separación entre circuitos y usuario/paciente	Separación entre etapa de control y potencia	Sí
Protección contra sobre corriente	Evitar daños por exceso de corriente	Protección integrada en driver TB6600	Sí
Protección térmica	Evitar sobrecalentamiento	Protección interna del driver	Sí
Funcionamiento esencial	Operación continua sin fallas críticas	Sistema controlado por microcontrolador	Sí
Condiciones de fallo	Seguridad ante fallos del sistema	No implementado completamente	No
Riesgos mecánicos	Protección ante partes móviles	No se cuenta con protección física completa	No
Diseño seguro	Minimización de riesgos para usuario	Diseño experimental sin certificación	Parcial
Ensayos y validación	Pruebas eléctricas normalizadas	No realizadas	No

Nota. Elaboración propia.

3.9 Criterios de selección de componentes electrónicos

La selección de los componentes electrónicos del prototipo se realizó considerando criterios técnicos, funcionales y de seguridad, con el objetivo de garantizar un desempeño adecuado del sistema y su integración eficiente dentro del diseño propuesto.

- ✓ **Compatibilidad eléctrica:** Se consideró la compatibilidad de niveles de tensión y

corriente entre los diferentes dispositivos, asegurando su correcto funcionamiento dentro de una arquitectura de alimentación multinivel (3.3 V, 5 V, 12 V y 24 V).

- ✓ Consumo energético: Se evaluó el consumo de corriente de cada componente con el fin de dimensionar adecuadamente la fuente de alimentación y evitar sobrecargas en el sistema.
- ✓ Funcionalidad y desempeño: Los componentes fueron seleccionados en función de su capacidad para cumplir con los requerimientos del sistema.
- ✓ Facilidad de integración: Se priorizó el uso de módulos compatibles entre sí, con interfaces estándar de comunicación (I2C, digital, serial), facilitando su implementación y programación.
- ✓ Seguridad: Se consideraron dispositivos que incorporan protecciones internas, tales como: Protección contra sobre corriente, protección térmica y regulación de voltaje.
- ✓ Disponibilidad y costo: Se seleccionaron componentes de fácil adquisición en el mercado local y de bajo costo, con el fin de garantizar la viabilidad del prototipo.
- ✓ Confiabilidad: Se eligieron componentes ampliamente utilizados y validados en aplicaciones similares, asegurando estabilidad y durabilidad en el funcionamiento del sistema.
- ✓ Adaptabilidad al prototipo: Los componentes fueron seleccionados considerando su facilidad de adaptación mecánica y electrónica al sistema, especialmente en el caso del motor, driver y sensores.

Tabla 18*Relación entre criterios de selección y componentes*

Componentes	Compatibilidad eléctrica	Bajo consumo	Funcionalidad	Integración	Seguridad	Disponibilidad	Confiabilidad
Raspberry Pi Pico (×2)	✓	✓	✓	✓	✓	✓	✓
Módulo Bluetooth	✓	✓	✓	✓	✓	✓	✓
ADS1115	✓	✓	✓	✓	✓	✓	✓
Sensor MPX5010 DP	✓	✓	✓	✓	✓	✓	✓
Pantalla OLED 128×64	✓	✓	✓	✓	✓	✓	✓
LED's	✓	✓	✓	✓	✓	✓	✓
Final de carrera	✓	✓	✓	✓	✓	✓	✓
Driver TB6600	✓	✗	✓	✓	✓	✓	✓
Motor NEMA 17	✓	✗	✓	✓	✓	✓	✓
Reguladores (5V / 3.3V)	✓	✓	✓	✓	✓	✓	✓
Fuente 12V / 24V	✓	✗	✓	✓	✓	✓	✓

Nota. Elaboración propia.

También se realizó una comparativa entre las posibles alternativas de cada uno de los componentes electrónicos.

- ✓ **Microcontrolador:** A continuación, se detallan las características de las posibles alternativas de microcontrolador y se selecciona la mejor de ellas.

Tabla 19*Características de las posibles alternativas de microcontrolador*

Parámetro	Raspberry Pi Pico	Arduino	ESP32
Voltaje entrada	1.8 – 5.5 V	7 – 12 V	3.0 – 3.6 V
Voltaje lógico	3.3 V	5 V	3.3 V
Corriente típica	20 – 30 mA	~50 mA	80 – 240 mA
Frecuencia CPU	133 MHz	16 MHz	240 MHz
Núcleos	2	1	2
RAM	264 KB	2 KB	~520 KB
ADC	12 bits	10 bits	12 bits
GPIO	26	14	30
Precio (\$)	15 – 25	40 – 60	25 – 40

Nota. Elaboración propia.**Tabla 20***Relación entre criterios de selección y las alternativas de microcontrolador*

Criterio	Raspberry Pi Pico	Arduino	ESP32
Compatibilidad eléctrica	Alto	Alto	Medio
Bajo consumo	Alto	Medio	Medio
Funcionalidad	Alto	Medio	Alto
Integración	Alto	Alto	Medio
Seguridad	Alto	Alto	Medio
Disponibilidad	Alto	Alto	Alto
Confiabilidad	Alto	Alto	Medio

Nota. Elaboración propia.

En cuanto al sistema de control, se seleccionó el microcontrolador Raspberry Pi Pico microcontroller debido a su adecuada capacidad de procesamiento (133 MHz), bajo consumo de corriente y arquitectura de doble núcleo, lo que permite ejecutar tareas de control y adquisición de datos de manera simultánea. Además, su compatibilidad con niveles de voltaje de 3.3 V y su disponibilidad en el mercado lo convierten en una opción eficiente frente a alternativas como Arduino Uno o ESP32.

✓ **Sensor de presión:**

Tabla 21

Características de las posibles alternativas de sensor de presión

Parámetro	MPX5010DP	MPX5050	Honeywell TruStability HSC
Voltaje	5 V	5 V	3.3 – 5 V
Corriente	~7 mA	~7 mA	10 – 15 mA
Rango presión	0 – 10 kPa (0–102 cmH ₂ O)	0 – 50 kPa	Variable
Sensibilidad	≈ 450 mV/kPa	menor	Alta
Precisión	±2.5%	±2.5%	±1%
Salida	Analógica	Analógica	Digital
Tiempo respuesta	<1 ms	<1 ms	<1 ms
Precio (S/)	55 – 95	75 – 115	150 – 300

Nota. Elaboración propia.

Tabla 22

Relación entre criterios de selección y las alternativas de sensor de presión

Criterio	MPX5010DP	MPX5050	Honeywell
Compatibilidad eléctrica	Alto	Alto	Alto
Bajo consumo	Alto	Alto	Medio
Funcionalidad	Alto	Medio	Alto
Integración	Alto	Alto	Medio
Seguridad	Alto	Medio	Alto
Disponibilidad	Alto	Medio	Bajo
Confiabilidad	Alto	Alto	Alto

Nota. Elaboración propia.

Para la medición de presión, se empleó el sensor MPX5010DP pressure sensor, el cual presenta un rango de operación de 0 a 10 kPa (equivalente a 0–102 cmH₂O), adecuado para aplicaciones de ventilación mecánica. Su salida analógica, bajo consumo de corriente (~7 mA) y precisión del ±2.5% permiten una implementación sencilla y confiable, sin necesidad de interfaces complejas, a diferencia de sensores digitales de mayor costo.

✓ **Conversor análogo digital:**

Tabla 23

Características de las posibles alternativas de conversor A/D

Parámetro	ADS1115	ADC interno Pico	MCP3008
Voltaje	2.0 – 5.5 V	3.3 V	2.7 – 5.5 V
Corriente	150 μ A	—	320 μ A
Resolución	16 bits	12 bits	10 bits
Canales	4	3	8
Ganancia programable	Sí	No	No
Frecuencia muestreo	860 SPS	Alta	200 kSPS
Interfaz	I2C	Interno	SPI
Precio (S/)	10 – 20	—	8 – 15

Nota. Elaboración propia.

Tabla 24

Relación entre criterios de selección y las alternativas de conversor A/D

Criterio	ADS1115	ADC interno Pico	MCP3008
Compatibilidad eléctrica	Alto	Alto	Alto
Bajo consumo	Alto	Alto	Alto
Funcionalidad	Alto	Medio	Bajo
Integración	Alto	Alto	Medio
Seguridad	Alto	Medio	Medio
Disponibilidad	Alto	Alto	Alto
Confiabilidad	Alto	Alto	Alto

Nota. Elaboración propia.

Con el fin de mejorar la resolución en la adquisición de señales, se incorporó el convertidor analógico-digital ADS1115 ADC module, el cual ofrece una resolución de 16 bits y una tasa de muestreo de hasta 860 muestras por segundo (SPS). Esto permite obtener mediciones más precisas de la presión en comparación con el ADC interno del microcontrolador, contribuyendo a un mejor desempeño del sistema de control.

✓ **Motor:**

Tabla 25

Características de las posibles alternativas de motor

Parámetro	NEMA 17	Motor DC	Servomotor SG90
Voltaje	12 V	6 – 12 V	5 – 7.4 V
Corriente	1 – 2 A	0.5 – 2 A	0.5 – 1 A
Potencia	~12 – 24 W	Variable	Baja
Torque	40–50 N·cm	Bajo	Medio
Precisión	1.8°/paso	Baja	Media
Control	Posición exacta	Velocidad	Posición
Precio (\$)	40 – 75	10 – 30	20 – 55

Nota. Elaboración propia.

Tabla 26

Relación entre criterios de selección y las alternativas de motor

Criterio	NEMA 17	Motor DC	Servomotor SG90
Compatibilidad eléctrica	Alto	Alto	Medio
Bajo consumo	Medio	Alto	Medio
Funcionalidad	Alto	Bajo	Medio
Integración	Alto	Medio	Medio
Seguridad	Alto	Bajo	Medio
Disponibilidad	Alto	Alto	Alto
Confiabilidad	Alto	Medio	Medio

Nota. Elaboración propia.

En el sistema de actuación mecánica, se seleccionó el motor NEMA 17 debido a su capacidad de proporcionar un control preciso de posición (1.8° por paso) y un torque suficiente para la compresión repetitiva de la bolsa AMBU. Esta característica resulta fundamental para garantizar un volumen de ventilación constante, lo cual no se logra con la misma precisión utilizando motores DC o servomotores.

✓ **Driver:**

Tabla 27

Características de las posibles alternativas de driver

Parámetro	TB6600	A4988	DRV8825
Voltaje	9 – 42 V	8 – 35 V	8.2 – 45 V
Corriente	4 A	2 A	2.5 A
Potencia	Alta	Media	Media
Protección térmica	Sí	Sí	Sí
Aislamiento	Sí	No	No
Precio (S/)	40 – 60	8 – 20	10 – 25

Tabla 28

Relación entre criterios de selección y las alternativas de driver

Criterio	TB6600	A4988	DRV8825
Compatibilidad eléctrica	Alto	Alto	Alto
Bajo consumo	Medio	Alto	Alto
Funcionalidad	Alto	Medio	Medio
Integración	Alto	Medio	Medio
Seguridad	Alto	Medio	Medio
Disponibilidad	Alto	Alto	Alto
Confiabilidad	Alto	Medio	Medio

Nota. Elaboración propia.

Mejor opción: TB6600, ya que su alimentación está en el rango del motor NEMA 17, también presenta protección térmica y aislamiento lo cual compensa su alto precio

El control del motor se realiza mediante el driver TB6600, el cual permite manejar corrientes de hasta 4 A y operar en un rango de voltaje amplio (9–42 V). Este driver proporciona mayor robustez, protección térmica y estabilidad en comparación con alternativas de menor capacidad, lo que resulta adecuado para aplicaciones de funcionamiento continuo.

✓ **Pantalla:**

Tabla 29

Características de las posibles alternativas de pantalla

Parámetro	OLED 128×64	LCD 16x2	TFT LCD display
Voltaje	3.3 – 5 V	5 V	3.3 – 5 V
Corriente	~20 mA	~60 mA	100–300 mA
Resolución	128×64	16x2	Alta
Contraste	Muy alto	Bajo	Medio
Consumo	Bajo	Medio	Alto
Tiempo respuesta	Rápido (SPI)	Medio	Medio
Precio (\$)	15 – 30	10 – 20	30 – 75

Nota. Elaboración propia.

Tabla 30

Relación entre criterios de selección y las alternativas de pantalla

Criterio	OLED 128×64	LCD 16x2	TFT LCD display
Compatibilidad eléctrica	Alto	Alto	Medio
Bajo consumo	Alto	Alto	Bajo
Funcionalidad	Alto	Bajo	Alto
Integración	Alto	Alto	Medio
Seguridad	Alto	Medio	Medio
Disponibilidad	Alto	Alto	Medio
Confiabilidad	Alto	Alto	Medio

Nota. Elaboración propia.

Para la visualización de parámetros, se eligió una pantalla OLED de 128×64, la cual ofrece bajo consumo energético (~20 mA), alto contraste y buena legibilidad. Estas características permiten mostrar información relevante del sistema, como presión y estado de operación, de manera clara y eficiente, superando las limitaciones de pantallas LCD convencionales. Adicionalmente, se incorporaron módulos complementarios como el sistema de comunicación Bluetooth, que permite la transmisión inalámbrica de datos, y reguladores de voltaje de 5 V y 3.3 V, necesarios para garantizar la correcta alimentación de los distintos componentes del sistema. Asimismo, el uso de una fuente de alimentación de 12–24 V asegura el suministro adecuado de energía.

CAPITULO IV

DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

A partir del capítulo anterior, donde se abordó el dimensionamiento de la estructura y la adquisición de componentes electrónicos, ahora es posible avanzar con el diseño y ensamblaje, tanto a nivel de hardware como de software.

La primera parte implica el diseño de la estructura y la simulación para determinar el tipo de filamento a utilizar en la impresión en 3D. La segunda parte de este capítulo involucra la integración de los componentes físicos para ensamblar la estructura del ventilador mecánico. Durante esta etapa, se ensamblan las piezas impresas previamente. La tercera parte consiste en poner en marcha los componentes electrónicos, tales como el motor, el microcontrolador, el actuador de final de carrera, el display, el sensor de presión y otros elementos. Se realiza la conexión y configuración adecuada de cada componente.

Finalmente, en este capítulo se aborda la programación de las funciones establecidas en los objetivos y alcances del proyecto. Se emplean lenguajes como Micropython en el entorno de desarrollo Thonny IDE, APP Inventor y Visual Basic para desarrollar la interfaz gráfica de usuario (GUI).

4.1 Diseño e impresión en 3D

Una vez establecida la mejor alternativa de solución en el capítulo III, se procedió al diseño detallado utilizando el software SolidWorks en 3D. Durante este proceso, se definieron las especificaciones y características de todos los elementos que componen la solución. Se realizaron cálculos matemáticos de cargas y esfuerzos para establecer las dimensiones del mecanismo, asegurando que el objeto cumpla con los requerimientos solicitados.

Además, se empleó la tecnología de impresión en 3D para materializar los

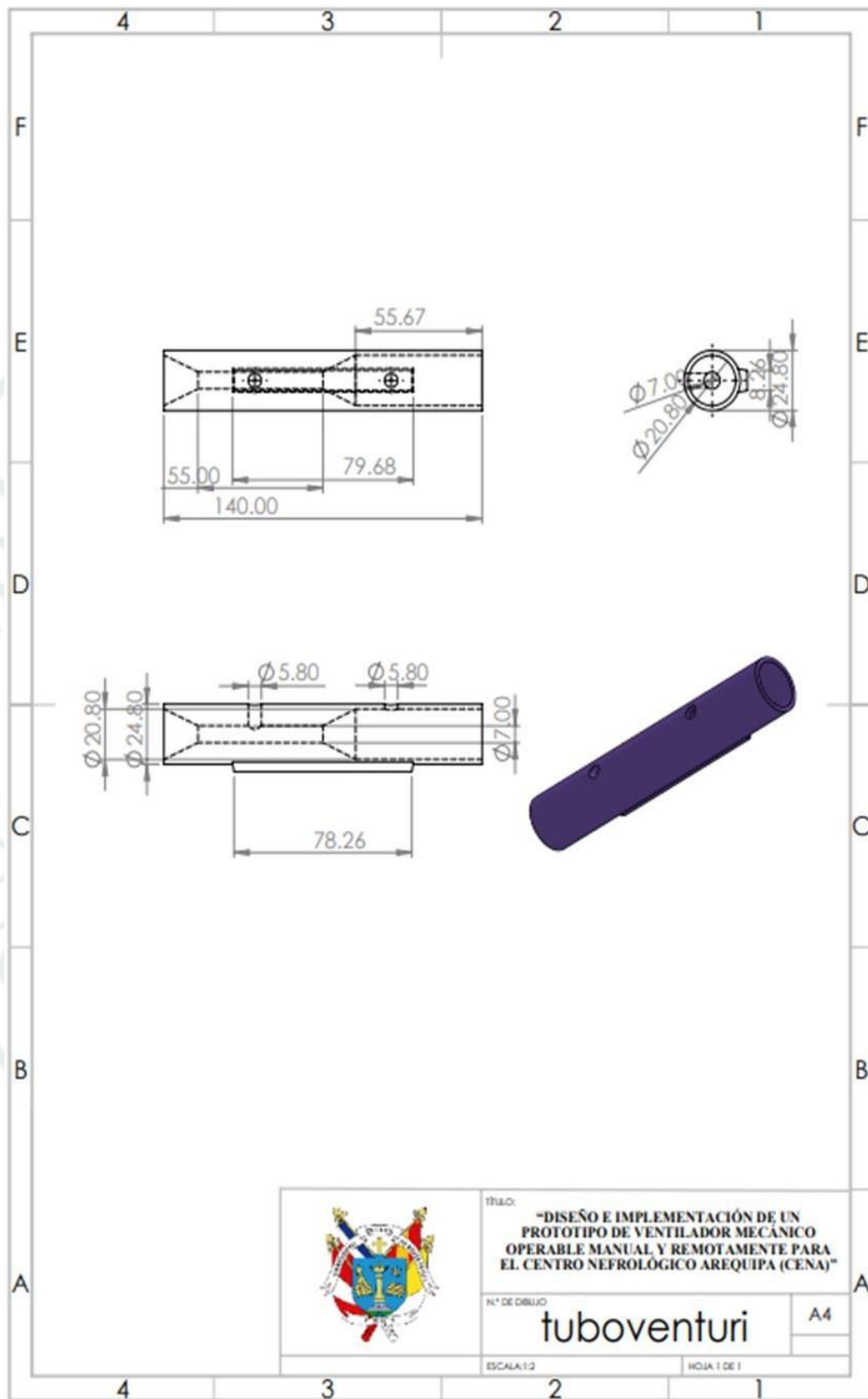
componentes diseñados, permitiendo una fabricación precisa y eficiente del dispositivo. Las ventajas de la impresión en 3D, como la capacidad de crear formas complejas y la reducción de costos y tiempos de producción, fueron aprovechadas en este proceso de diseño y fabricación.

4.1.1 Tubo de Venturi

Se diseñó utilizando los datos del dimensionamiento en el software SolidWorks. Es importante destacar que el tubo de Venturi tiene una longitud de 140 mm, con una abertura mayor (P1) de 20.8 mm de diámetro y una abertura menor (P2) de 7 mm. Además, se implementaron cortes diagonales para evitar la obstrucción del flujo de aire, siguiendo las recomendaciones de referencia (Kshetry et al., 2021). También se incorporaron dos boquillas para permitir el uso de un sensor de presión diferencial, como se ilustra en la Figura 46.

Figura 46

Plano y vistas del tubo de Venturi



Nota. Elaboración propia.

La pieza diseñada previamente en SolidWorks se procedió a imprimir en un lapso de 3 horas, tal como se muestra en la Figura 47. Para este proceso de impresión, se empleó filamento ABS de 1.75 mm de espesor, con una temperatura de fusión de 260°C.

Figura 47
Impresión del tubo de Venturi



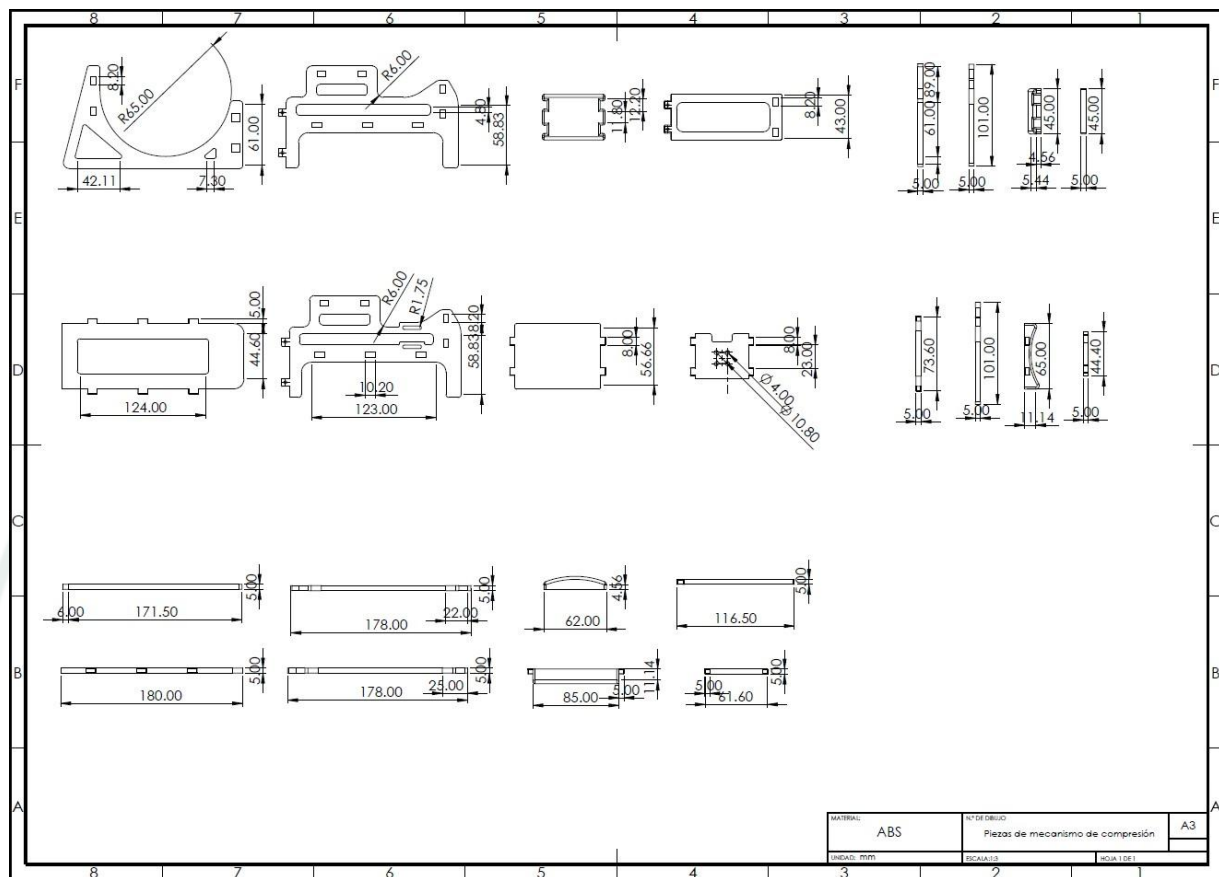
Nota. Elaboración propia.

4.1.2 Impulsor de Aire

A partir del criterio de la estructura del impulsor de aire y de la referencia de la tesis de Carpio y Sughey, (2022) se realiza un diseño en SolidWorks y un estudio mecánico de tensión a la que estará sometida la estructura para la selección del material con que debe de ser impreso.

Primeramente, se realiza el diseño de las piezas que conformarán el impulsor de aire, en la figura 48 se muestra las medidas de cada una de ellas a una escala de de 1:3 del tamaño original.

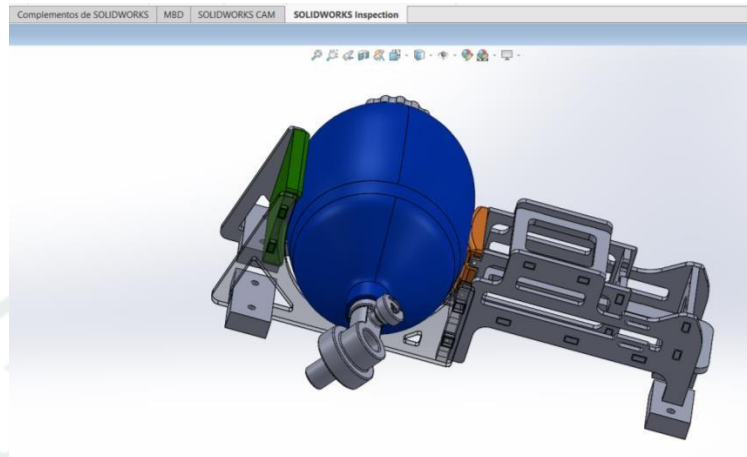
Figura 48
Plano de las piezas del impulsor de aire



Nota. Elaboración propia

Con las piezas disponibles, es posible realizar el ensamblaje en SolidWorks, como se ilustra en la Figura 49.

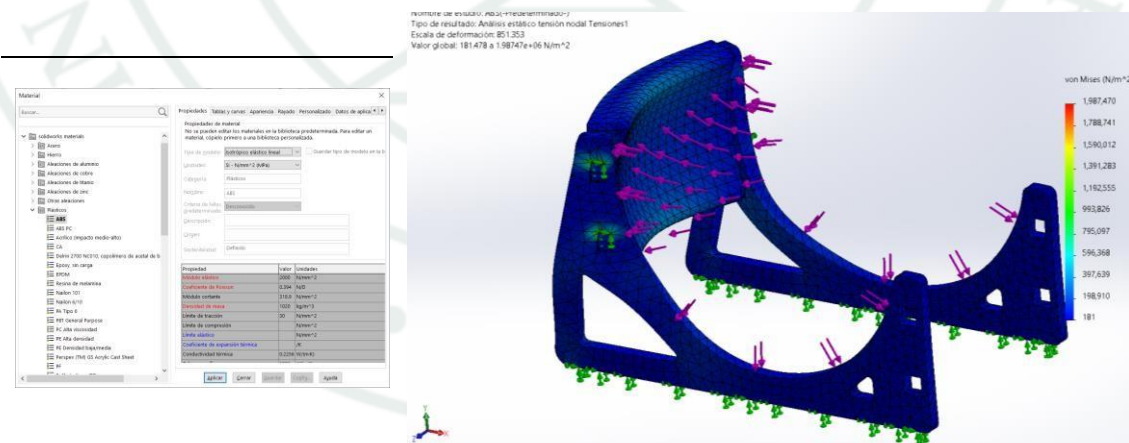
Figura 49
Estructura de la presión de la bolsa de Ambu

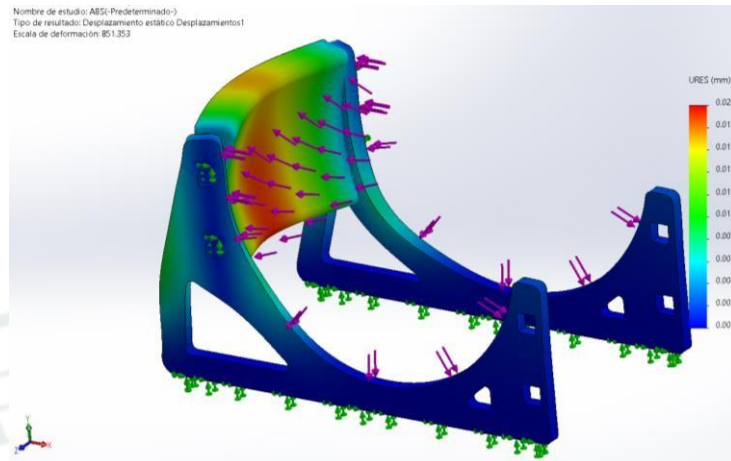
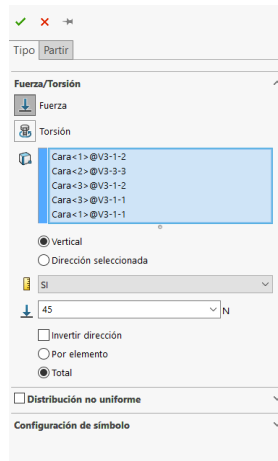


Nota. Elaboración propia

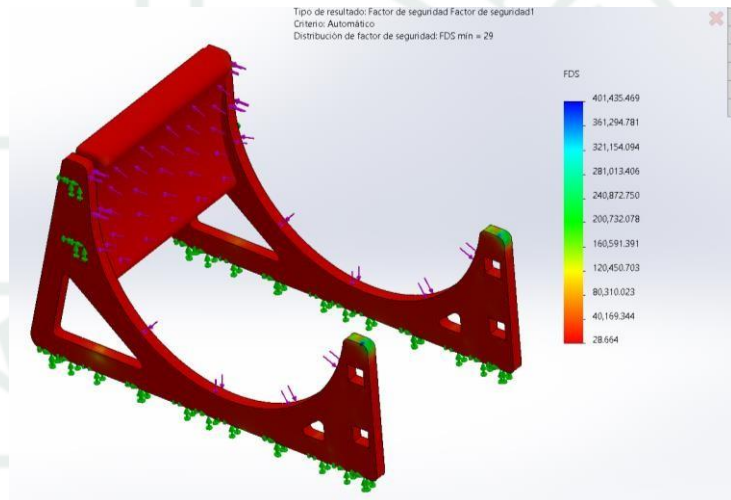
A continuación, se realizan pruebas estáticas para seleccionar el material más adecuado. Se consideran filamentos de ABS, PETG y Acrílico. En la siguiente tabla se comparan estos tres materiales y se busca aquel que presente un factor de seguridad al menos tres veces mayor y sufra la menor deformación bajo una fuerza de 45N, como se mencionó en el capítulo 3.

Tabla 19
Propuesta de materiales del prototipo ventilador mecánico
Material: ABS

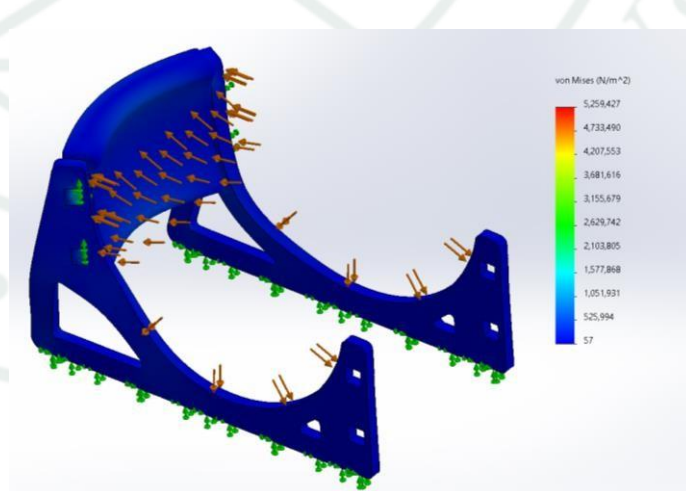
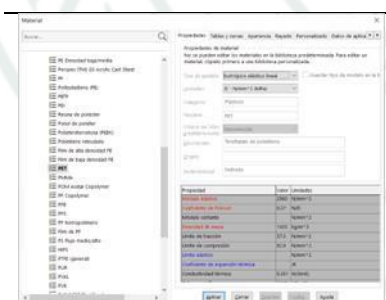




Nota: Se aplica una fuerza total de 45N. Como resultado de la simulación se tiene un Factor de Seguridad de 29, con una deformación de 0.021mm desde 181 N/m².

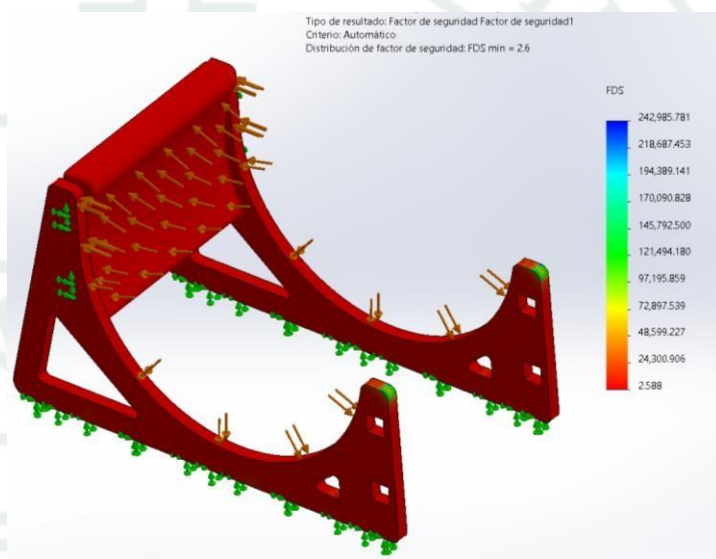
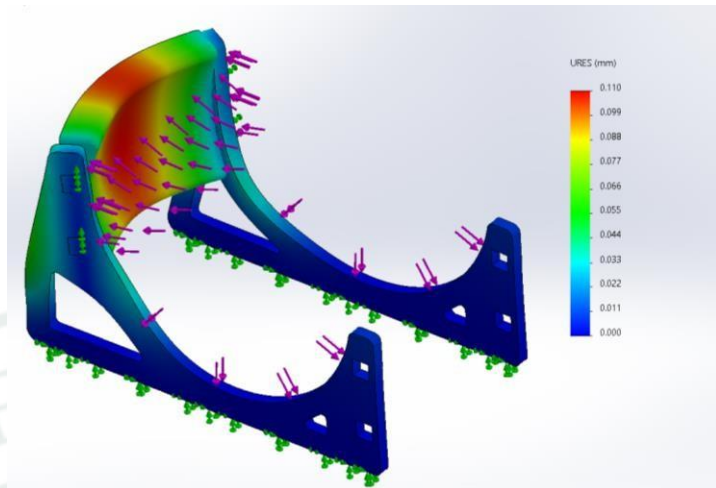


PET

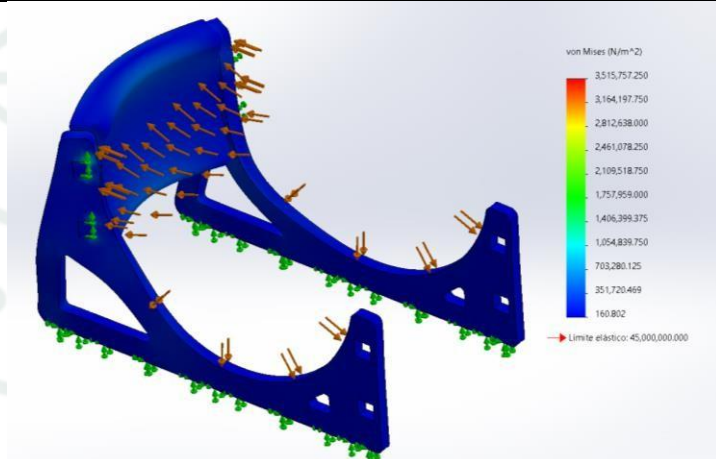


Nota: Se aplica una fuerza total de 45N. Como resultado de la simulación se tiene un Factor de Seguridad

de 2.6, con una deformación
de 0.11 mm desde 57 N/m2.

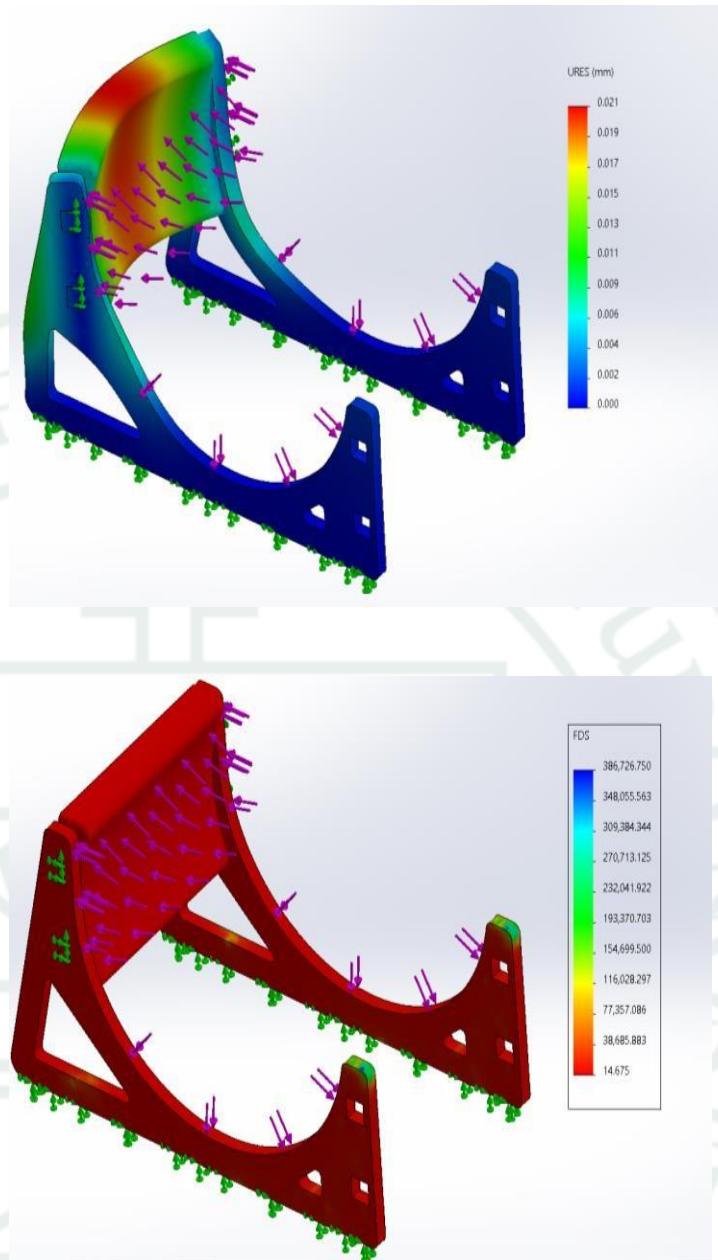


ACRILICO



Nota: Se aplica una fuerza
total de 45N. Como
resultado de la simulación se

tiene un Factor de Seguridad de 15, con una deformación de 0.021mm desde 160 N/m².



Nota. Elaboración propia.

Después de completar la simulación estática en SolidWorks, se ha llegado a la conclusión de que utilizar filamento ABS es la elección más apropiada. Esta decisión se fundamenta en su mínima variación de deformación, medida en 0.021 mm, y en su alto factor de seguridad de 29, lo que garantiza un funcionamiento seguro y fiable del prototipo.

Se procede con la impresión de las piezas a partir del diseño de la Estructura del Impulsor de Aire. El extenso conjunto de piezas 3D se detalla en el Anexo C4: Estructura de Compresión Explosionada, para la visualización completa. Se empleó filamento ABS de 1.75 mm de espesor para la impresión de estas piezas, como se muestra en la Figura 50.

Figura 50
Impresión de la pieza V3-1

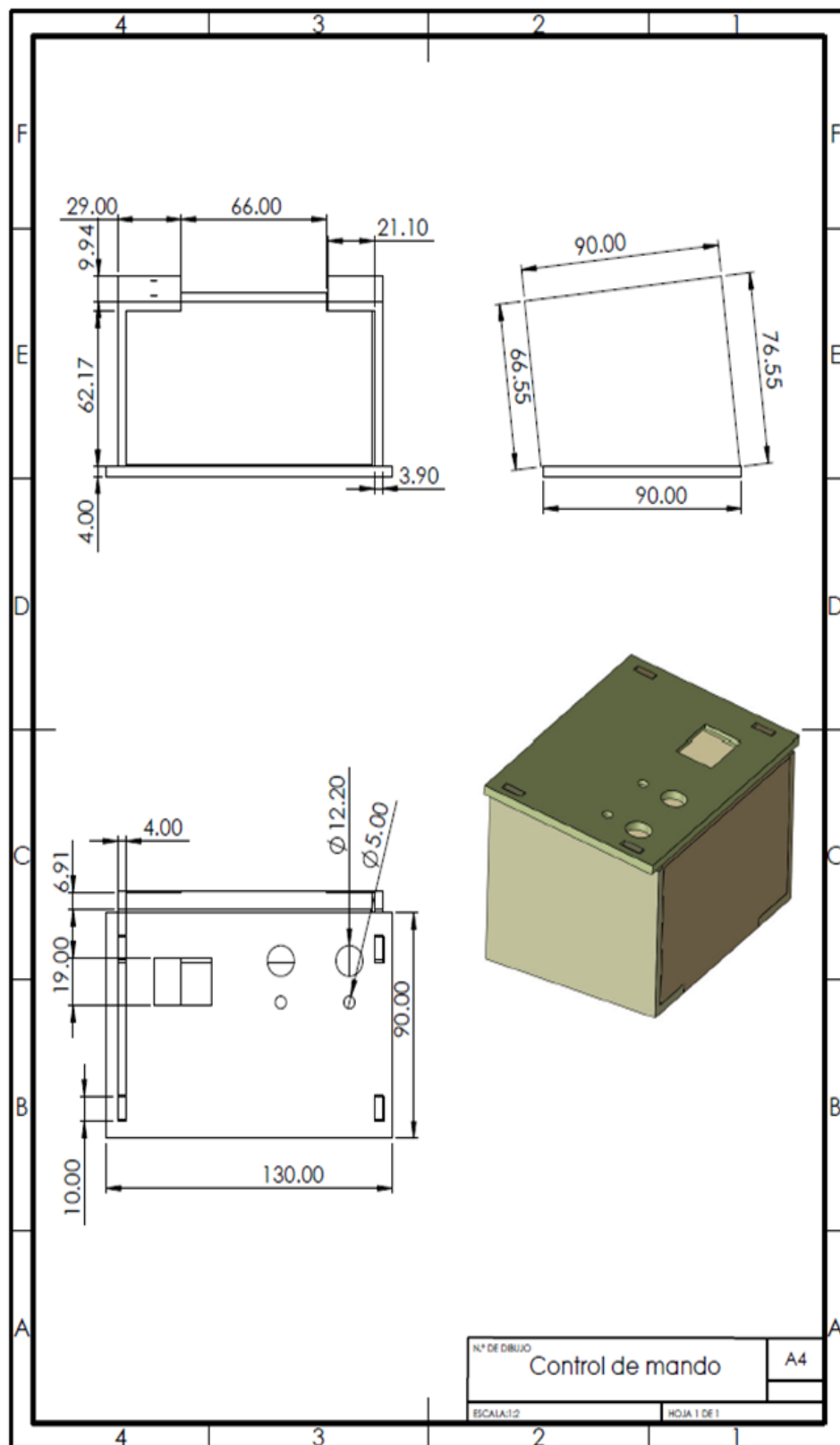


Nota. Fotografía tomada por los autores.

4.1.3 Control de mando

Las piezas de control de mando forman parte integral de la carcasa de protección del circuito. En la Figura 51 se presenta el plano que muestra las tres vistas laterales de estas piezas. Para obtener información más detallada sobre los planos, se puede consultar el Anexo C.

Figura 51
Plano y vistas del control de mando



Nota. Elaboración propia.

Se procede con la impresión de las piezas a partir del diseño de la Estructura del Control de mando. El extenso conjunto de piezas 3D se detalla en el Anexo C4: Estructura de Compresión Explosionada, para la visualización completa. Se empleó filamento ABS de 1.75 mm de espesor para la impresión de estas piezas, como se muestra en la Figura 52.

Figura 52
Piezas del control de mando

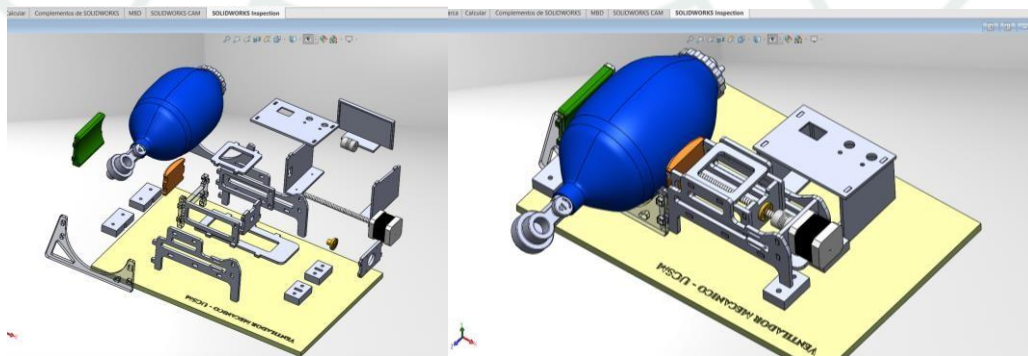


Nota. Elaboración propia

4.2 Implementación Estructural del Prototipo

Una vez realizado los cálculos, diseño, modelamiento y la impresión en 3D, se procede con la implementación de la estructura del ventilador mecánico. En la Figura 53 se muestra el modelamiento del ventilador en el software de SolidWorks, y en la parte derecha se muestra implementado con los componentes del prototipo.

Figura 53
Ensamblado 3D del prototipo en SolidWorks

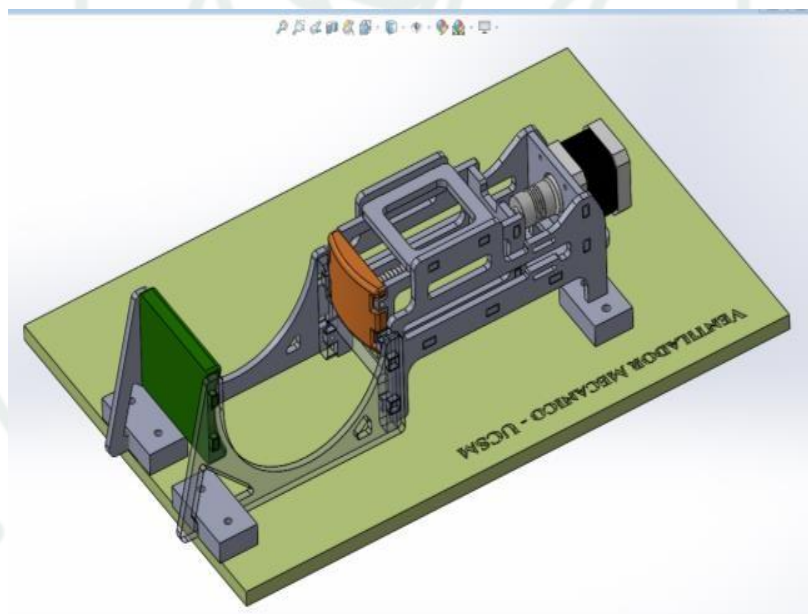


Nota. Elaboración propia.

4.2.1 *Extractor de Aire, Bolsa Ambu*

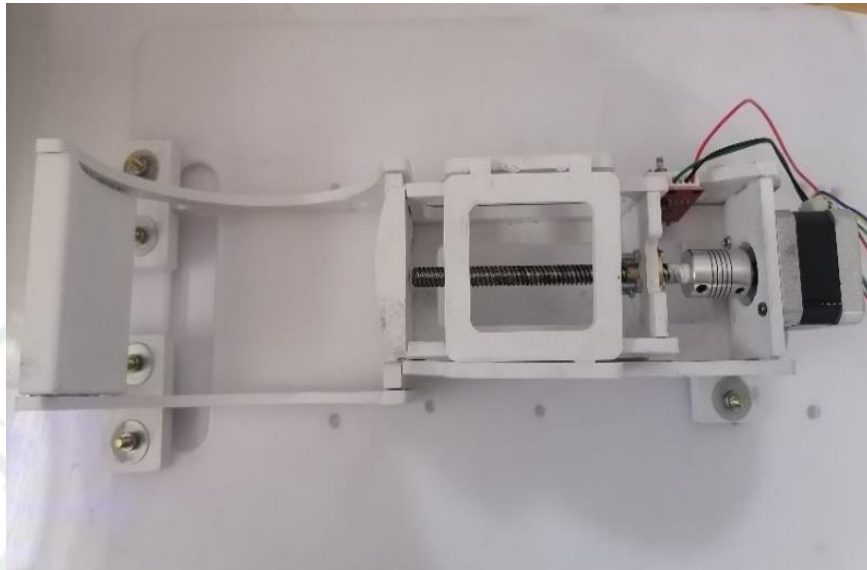
El extractor de aire a través de la bolsa Ambu, es el mecanismo principal de la ventilación, ya que simula la función que realiza una persona con las manos, por ende, tiene la contextura rígida y firme. En la Figura 54 se muestra el diseño previo en SolidWorks y en la Figura 55 se muestra ya montado; esta estructura se aseguró con pernos, ya que tiene que ser fija y no dañar el mismo motor por fenómenos de fricción y vibraciones, también es fácil de transportar y con una estructura pequeña en comparación de las otras investigaciones que se mencionan en los antecedentes.

Figura 54
Extractor de aire diseñado en SolidWorks



Nota. Elaboración propia.

Figura 55
Extractor de aire montado



Nota. Elaboración propia.

Tabla 20
Características del extractor de aire

Característica del extractor de aire	Descripción
Tamaño	40 cm de largo, 13 cm de altura y 9 cm de profundidad
Color	Blanco
Material	ABS
Peso	850 gr. Sin el motor
Transporte	Tiene un aspecto compacto para la facilitar su movilidad.

Nota. Elaboración propia.

4.2.2 Control de Mando

El control de mando, posee a los botones del control de modos de operación y a su vez, también contiene al display que se encarga de mostrar y el caudal de aire expulsa el ventilador mecánico. Dentro del control de mando se incrustará el circuito que contiene al microcontrolador. Las dimensiones que posee son de 11 cm largo, 9 cm altura y 12 cm de profundidad (Figura 56).

Figura 56
Control de mando



Nota. Elaboración propia.

4.2.3 Tubo de Venturi

Una vez realizado la impresión en 3D, se procede con la integración al prototipo. Para ello se realizó el empalme del tubo de Venturi con el tubo transparentes de 1/8 (delgado), en los dos orificios como se muestra en la parte izquierda de la Figura 57 y posteriormente en el tubo de 1 ½ respetando el teorema de Bernoulli, el diámetro mayor hacia el suministro de aire como se muestra en la parte derecha de la Figura 57.

Figura 57
Empalme del tubo de Venturi



Nota. Elaboración propia.

4.3 Implementación Eléctrica, Electrónica del Prototipo

En esta etapa se lleva a cabo el comisionamiento de los dispositivos electrónicos necesarios para el control del prototipo del ventilador mecánico. El foco principal está en la distribución de los puertos de los microcontroladores que serán utilizados para conectar los sensores, módulos, drivers y actuadores. Se consideran específicamente los puertos de la Raspberry Pi Pico, siguiendo las indicaciones del fabricante. Para esto, se elaboró el Diagrama 2 con la asistencia del software Fritzing. Este diagrama detalla los componentes necesarios y busca optimizar la distribución de los pines. En él, los 2 Raspberry Pi Pico gestionan tanto los sensores como los actuadores. El primero se encarga del control del motor paso a paso a través del driver, que a su vez es controlado por el sensor y los pulsadores. Además, incluye indicadores como los Leds. Por otro lado, el segundo microcontrolador, ubicado en la parte derecha del diagrama, se encarga de la comunicación, tanto alámbrica como inalámbrica, y sirve como interfaz de usuario a través del display OLED.

Por otra parte, se desarrolló el esquema eléctrico para conectar las entradas y salidas de los puertos del mismo microcontrolador, junto con los módulos como Bluetooth, ADS1115, sensor MPX5010, display OLED y otros, detallados en el diagrama 3. Este esquema será fundamental para la posterior creación del circuito PCB. El software EasyEDA ha integrado algunos de estos componentes conforme a las especificaciones del fabricante. Aquellos que no estaban disponibles se sustituyeron por similares, asegurando que las dimensiones se mantengan constantes al ensamblar o soldar los componentes electrónicos en la placa PCB.

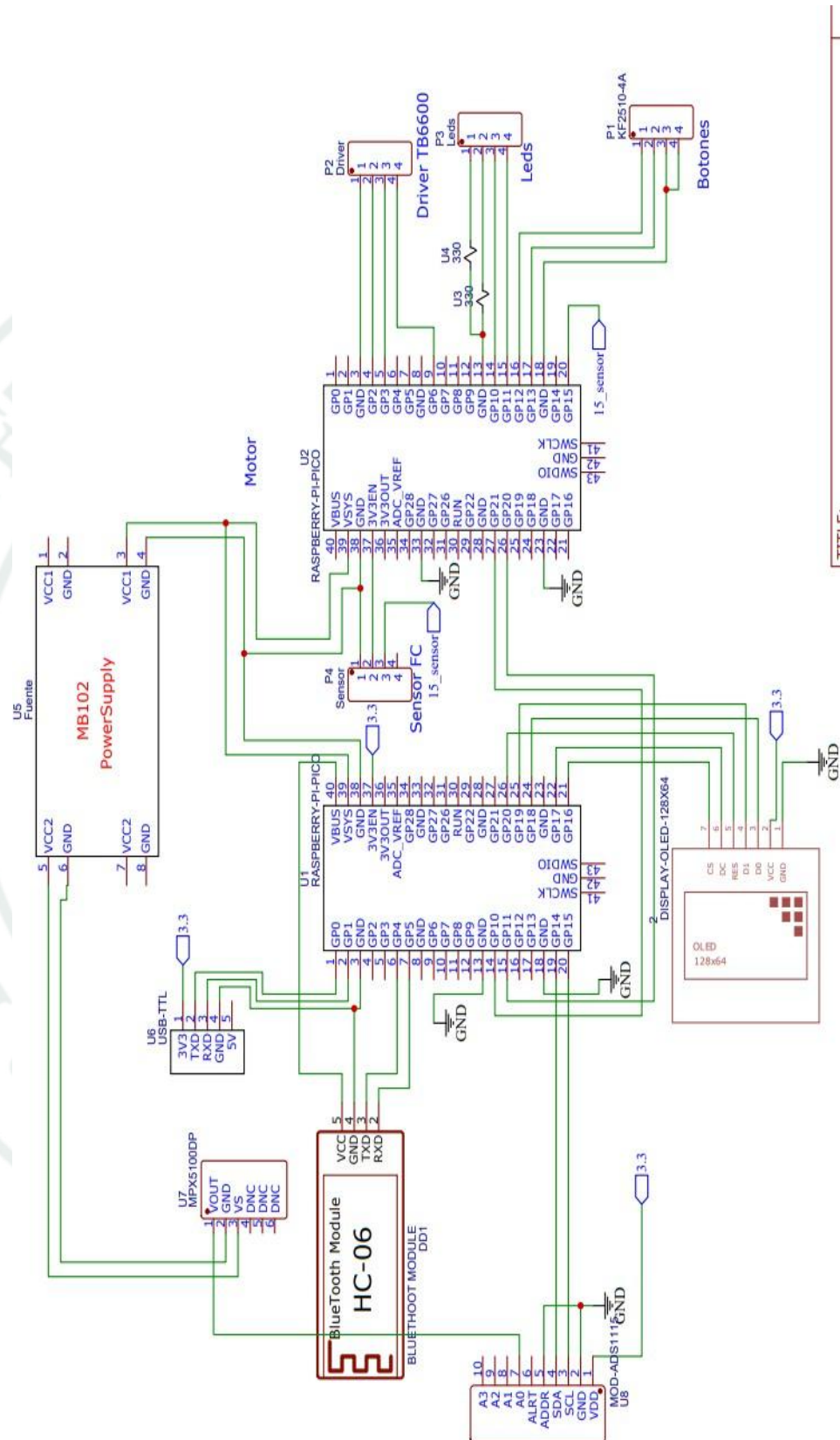
Esquemático de los componentes electrónicos



109

Figura 59

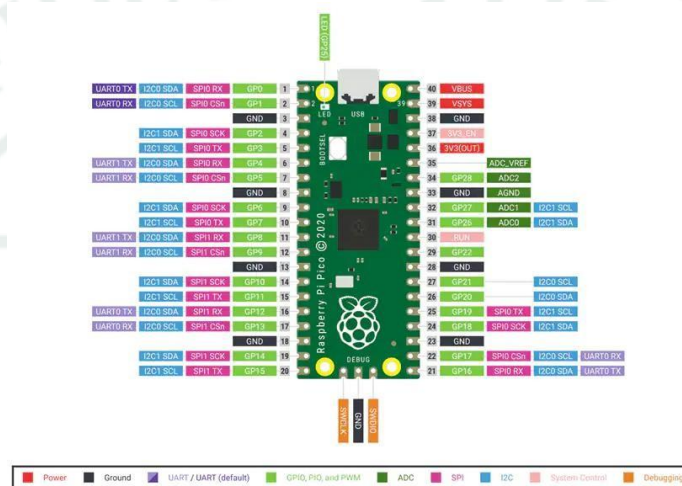
Esquema electrónico de conexión y alimentación para el prototipo



A continuación, se describe la implementación de la parte de instrumentación (eléctrica) tomando de guía el Figura 59 y la distribución de pines del microcontrolador Raspberry Pi Pico R3, mayor detalle en el Anexo E. En la Figura 60, se muestra los puertos de salidas y entradas del microcontrolador.

Figura 60

Distribución de pines de Raspberry Pi Pico

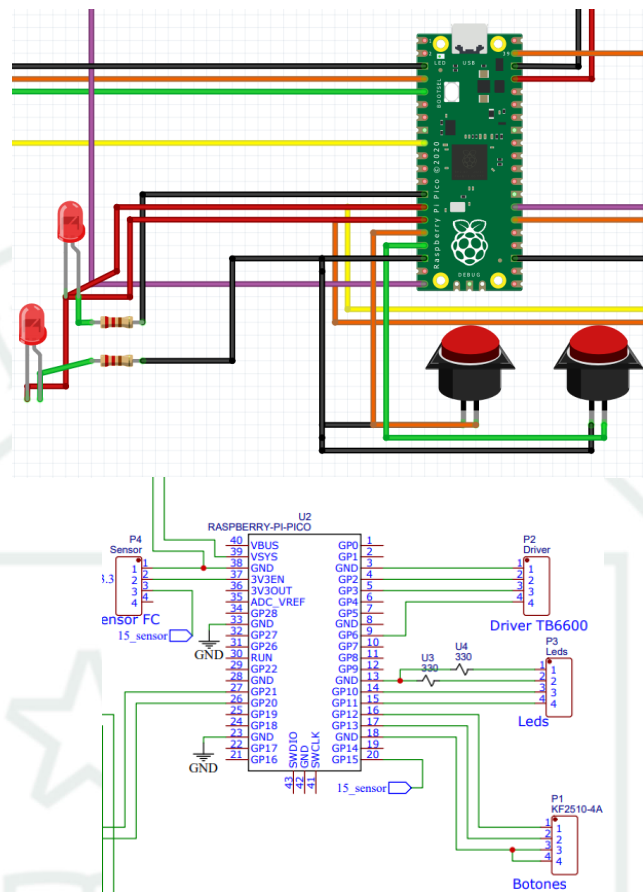


Nota. Raspberrypi.com 16 60

4.3.1 Local o Manual

El modo local o manual está conformado por el microcontrolador Raspberry Pi Pico 1, al cual se asignó 2 pines en los puertos de 12 y 13 para la función de los 2 botones pulsadores, seguidamente 2 diodos led de color rojo y azul en los pines 10 y 11 respectivamente, como se muestra en la Figura 61.

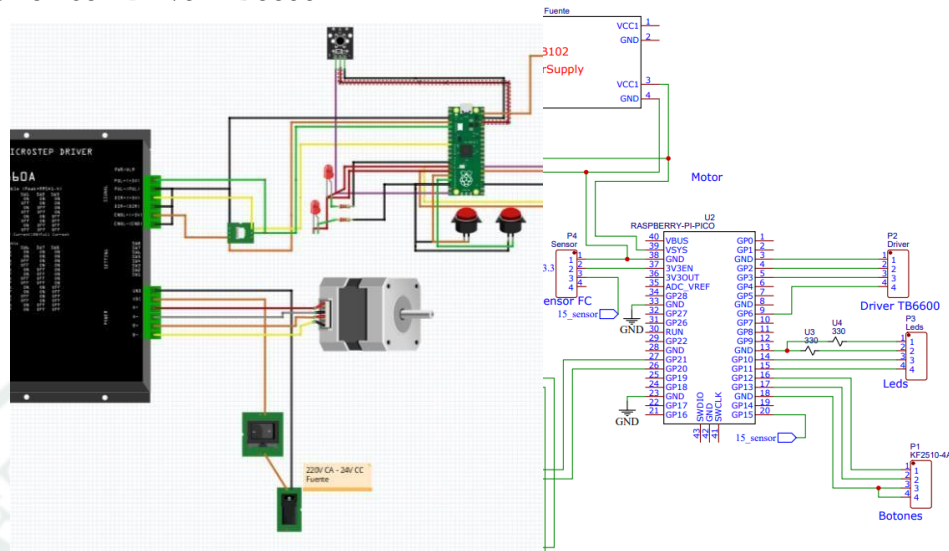
Figura 61
Conexión para modo manual



Nota. Elaboración propia.

Asimismo, se realizó la conexión con el Driver TB6600 como se muestra en la Figura 62, donde los 3 puertos de polaridad nengativa (- PUL , -DIR y -ENABLE) van conectados al GND del microcontrolador. Y los puertos de polaridad positiva del Driver (PUL+, DIR+ y Enable) se conectaron a los pines (3, 6 y 2) , por otra parte al sensor de final de carrera se le asigno el PIN 15.

Figura 62
Conexión con Driver TB6600

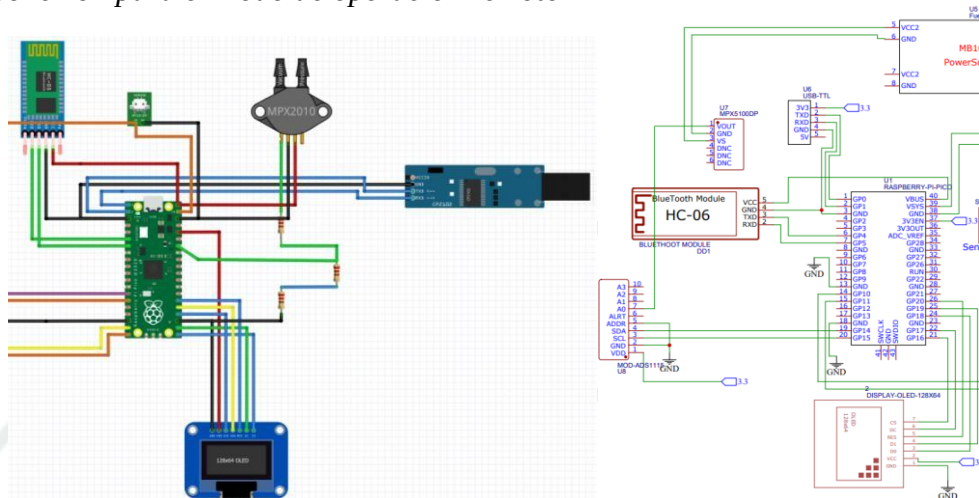


Nota. Elaboración propia.

4.3.2 Modo Remoto

El modo remoto está conformado por el microcontrolador Raspberry Pi Pico 2, como se puede ver en la Figura 63, donde los puertos GP0, GP1, GP4 y GP5 están conectados con los módulos de comunicación alámbrica (modulo serial TTL) e inalámbrica (Modulo Bluetooth CH-06), a través de la configuración UARTX0, UARDX0, UARCTX1 y UARCTX1. También en el esquema de la Figura 63, se le agrega el sensor de presión diferencial MPX5010DP usando la configuración de divisor de voltaje, tomando en cuenta que el sensor trabaja con 5V y el microcontrolador recibe señales analógicas de 3.3V. Y el *display* OLED 128x64 está conectado a los puertos mediante el protocolo de comunicación SPI.

Figura 63
Conexión para el modo de operación remoto



Nota. Elaboración propia.

4.4 Programación de los Microcontroladores con Micropython

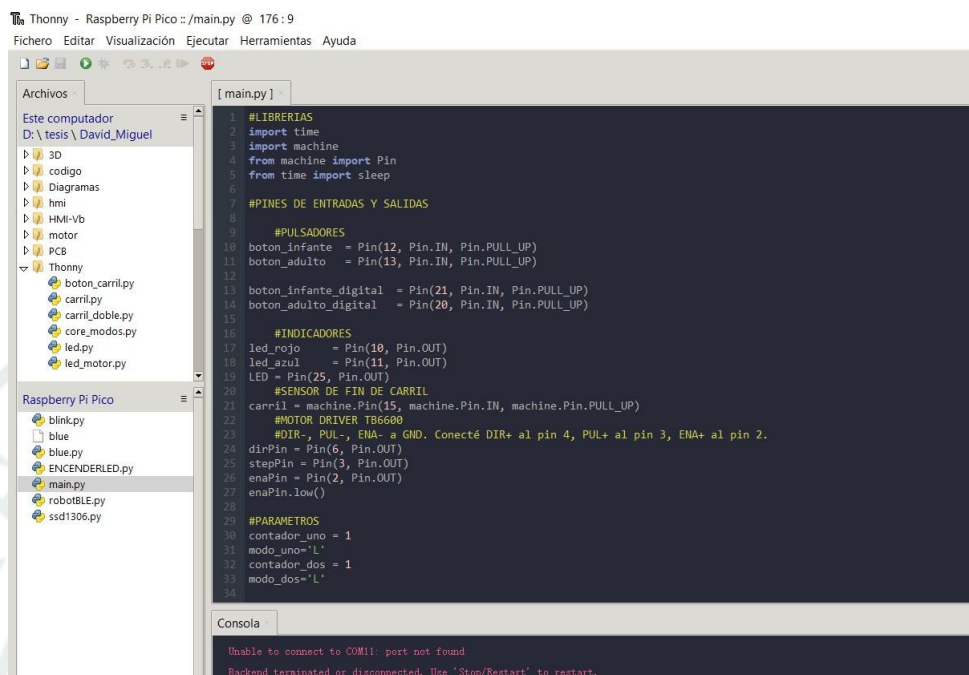
En esta sección, se describe el código de programación en base a los diagramas de flujo y se elaboraron los códigos utilizando el entorno de desarrollo integrado (IDE) Thonny, utilizando el lenguaje Micropython para ambos microcontroladores, tal como se muestra en la Figura 64.

Thonny, en contraste con el Arduino comúnmente empleado en investigaciones similares, destaca por su ligereza. La complejidad al trabajar en el entorno o lenguaje de los microcontroladores radica en la obtención de librerías y funciones para ciertos dispositivos electrónicos. La estructura de programación se asemeja al lenguaje nativo de Python, combinando elementos de C del conocido entorno de Arduino. Esto implica mantener el orden de las líneas de código de manera similar.

La compilación y la carga del código hacia el microcontrolador son casi instantáneas, con una latencia notablemente menor en comparación con el entorno de Arduino. Esta característica resulta útil para la modificación ágil del programa (código de programación).

Figura 64

Captura de pantalla de la programación en IDE de Thonny



```
Thonny - Raspberry Pi Pico - /main.py @ 176:9
Fichero  Editar  Visualización  Ejecutar  Herramientas  Ayuda

Archivos
Este computador
D:\tesis\David_Miguel
  3D
  codigo
  Diagramas
  hmi
  HMI-Vb
  motor
  PCB
  Thonny
    boton_carril.py
    carril.py
    carril_doble.py
    core_modos.py
    led.py
    led_motor.py
Raspberry Pi Pico
  blink.py
  blue
  blue.py
  ENCENDERLED.py
  main.py
  robotBLE.py
  ssd1306.py

[main.py]
1 #LIBRERIAS
2 import time
3 import machine
4 from machine import Pin
5 from time import sleep
6
7 #PINES DE ENTRADAS Y SALIDAS
8
9 #PULSADORES
10 boton_infante = Pin(12, Pin.IN, Pin.PULL_UP)
11 boton_adulto = Pin(13, Pin.IN, Pin.PULL_UP)
12
13 boton_infante_digital = Pin(21, Pin.IN, Pin.PULL_UP)
14 boton_adulto_digital = Pin(20, Pin.IN, Pin.PULL_UP)
15
16 #INDICADORES
17 led_rojo = Pin(10, Pin.OUT)
18 led_azul = Pin(11, Pin.OUT)
19 LED = Pin(25, Pin.OUT)
20
21 #SENSOR DE FIN DE CARRIL
22 carril = machine.Pin(15, machine.Pin.IN, machine.Pin.PULL_UP)
23
24 #MOTOR DRIVER TB6600
25 #DIR-, PUL-, ENA- a GND. Conecté DIR+ al pin 4, PUL+ al pin 3, ENA+ al pin 2.
26 dirPin = Pin(6, Pin.OUT)
27 stepPin = Pin(3, Pin.OUT)
28 enaPin = Pin(2, Pin.OUT)
29 enaPin.low()
30
31 #PARAMETROS
32 contador_uno = 1
33 modo_uno = 'L'
34 contador_dos = 1
35 modo_dos = 'L'
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Consola
Unable to connect to COM11: port not found
Backend terminated or disconnected. Use 'Stop/Restart' to restart.
```

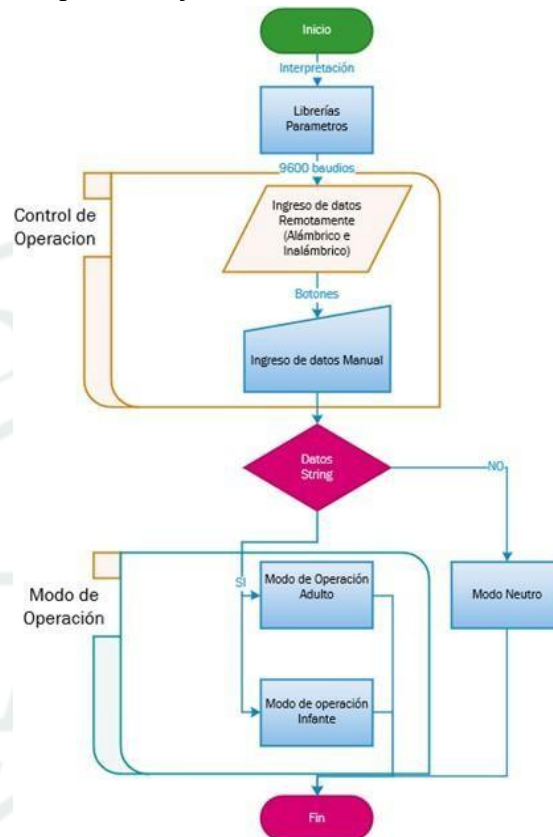
Nota. Elaboración propia

Antes de iniciar la programación, se establece un diagrama de flujo, visible en la figura 65. El propósito es definir los modos de control (local y remoto) y los modos de operación (adulto e infante).

El proceso se inicia con la interpretación de las librerías y los parámetros necesarios para los microcontroladores, utilizando un ciclo de 9600 baudios. Luego, se procede al ingreso de datos de control a través de botones físicos y virtuales. Estos datos se emplean para seleccionar los modos de operación. Si se elige el modo infante o adulto, el ciclo se completa. Si se queda en un estado neutro, la operación finaliza con el apagado del servomotor.

Figura 65

Diagrama de flujo de la operación y control de los microcontroladores



Nota. Elaboración propia.

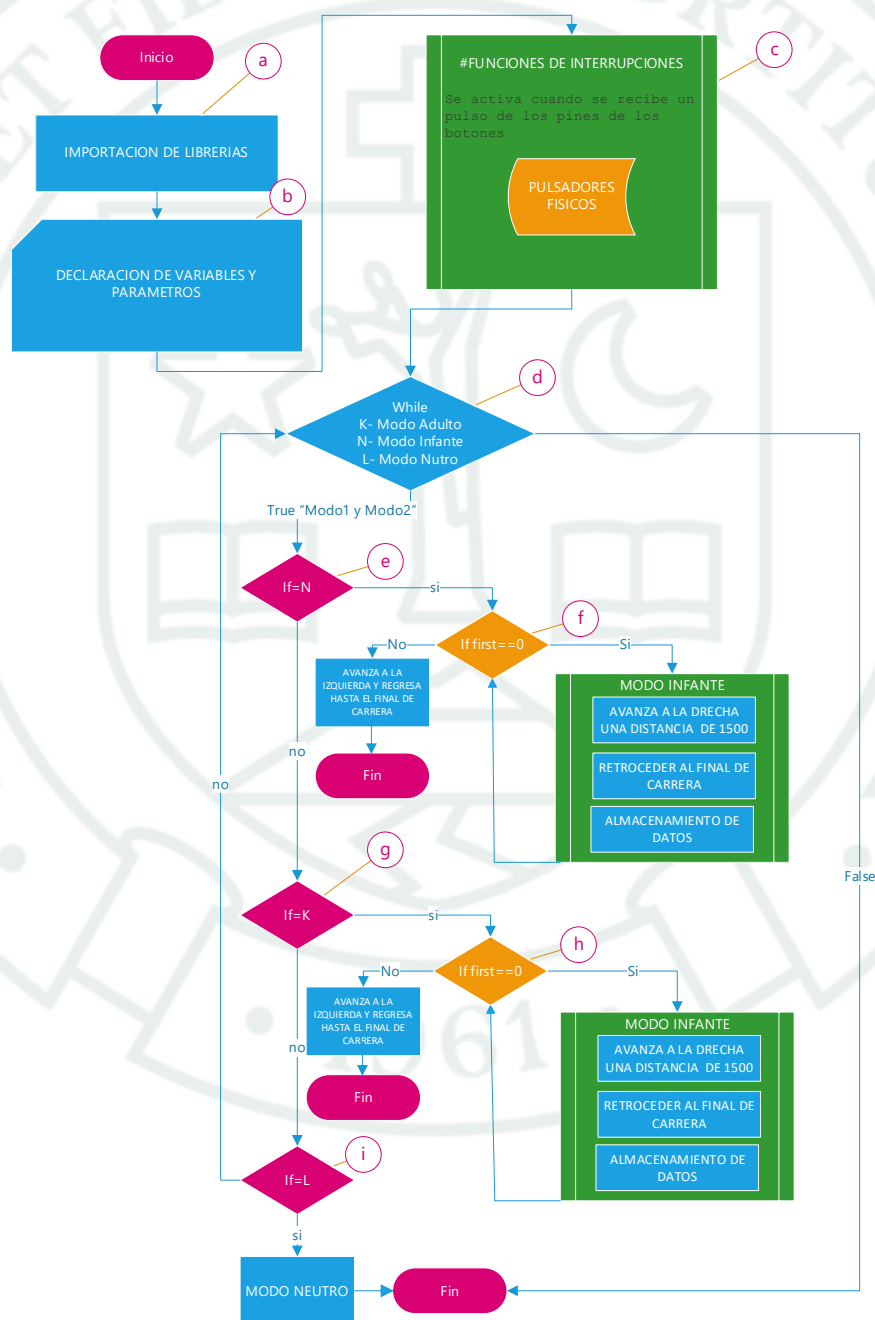
4.4.1 Control Manual o Local

A continuación, se presenta la Figura 66, que ilustra el flujo de operación para el control manual. Este proceso comienza con la importación de librerías y paquetes predefinidos por el entorno de desarrollo Thonny, seguido por la declaración de parámetros y variables. Luego, se crean dos funciones de interrupción asociadas a los pulsadores físicos y digitales. Y mientras haya una orden activada por los pulsadores, la función se activará según la condición. Si la orden es 'L', se activa el modo infante con una distancia de recorrido y velocidad predefinidas. Si la orden es 'K', se activa el modo adulto, permitiendo ajustes independientes de la distancia y la velocidad del motor. Finalmente, si la condición del bucle 'if' es 'N', se activa el modo neutro. El proceso vuelve

al bucle 'while' hasta recibir una nueva orden; en ausencia de respuesta, se finaliza el proceso.

El código se desarrolló en Micropython, como se detalla en el Anexo F1: Micropython R1.

Figura 66
Diagrama de flujo, modo manual



Nota. Elaboración propia.

Programación

La programación del microcontrolador para el control manual del ventilador mecánico del Anexo F1: Micropython R1, se basa en la secuencia lógica de la Figura 66, que exhibe una serie de burbujas numeradas alfabéticamente desde la 'a' hasta la 'h'.

- a. En primer lugar, se importaron las librerías, tal como se observa en la Figura 67, desde la fila 2 hasta la 5, donde se incluyen las librerías de tiempo, pines y sleep.

Figura 67

Importación de librerías, Micropython

```
1  #LIBRERIAS
2  import time
3  import machine
4  from machine import Pin
5  from time import sleep
```

Nota. Elaboración propia.

- b. A continuación, se procede a la declaración de variables y parámetros en el código, abarcando desde la fila 10 hasta la 33 del Anexo F1. En esta sección, se realiza la configuración de los pulsadores con una configuración PULL. UP, se definen los 2 LEDs como indicadores y se configura el LED interno del microprocesador de Raspberry Pi Pico en modo de salida. Además, se detallan las variables de modulación por PWM para el Driver TB6600.

Figura 68

Declaración de variables, Micropython

```
9  #PULSADORES
10 boton_infante = Pin(12, Pin.IN, Pin.PULL_UP)
11 boton_adulto = Pin(13, Pin.IN, Pin.PULL_UP)
12
13 boton_infante_digital = Pin(21, Pin.IN, Pin.PULL_UP)
14 boton_adulto_digital = Pin(20, Pin.IN, Pin.PULL_UP)
15
16 #INDICADORES
17 led_rojo = Pin(10, Pin.OUT)
18 led_azul = Pin(11, Pin.OUT)
19 LED = Pin(25, Pin.OUT)
20 #SENSOR DE FIN DE CARRIL
21 carril = machine.Pin(15, machine.Pin.IN, machine.Pin.PULL_UP)
22 #MOTOR DRIVER TB6600
```

Nota. Elaboración propia.

- c. Se presentan las cuatro funciones de interrupción que van desde la fila 38 hasta la fila 98. Dos de ellas se asignan a los pulsadores físicos y las otras dos a los pulsadores digitales. Estas funciones incluyen contadores que responden al flanco de disparo o subida. De esta manera, se establecen dos estados: 'K' para encender el LED o 'L' para apagarlo.

Figura 69

Funciones de interrupción, Micropython

```
Ask EasyCode | Explain | Refactor
38 def boton_uno_interrupt_digital(pin):
39     global contador_uno
40     print (contador_uno)
41     contador_uno +=1
42     if contador_uno > 2:
43         contador_uno =1
44     if contador_uno ==2:
45         led_azul.on()
46         global modo_uno
47         modo_uno = 'K'
48     else:
49         led_azul.off()
50         modo_uno='L'
51 boton_infante_digital.irq(handler = boton_uno_interrupt_digital, trigger=Pin.IRQ_FALL
86 print (contador_dos)
87 contador_dos +=1
88 if contador_dos > 2:
89     contador_dos =1
90 if contador_dos ==2:
91     led_rojo.on() # Encender led
92     global modo_dos
93     modo_dos = 'N' #Guardar variable
94     #print(modo_dos)
95 else:
96     led_rojo.off()
97     modo_dos='L' # Apagar led y guardar variable
98 boton_adulto.irq(handler = boton_dos_interrupt, trigger=Pin.IRQ_FALLING)
```

Nota. Elaboración propia.

- d. El bucle “while True”, es el principal bucle infinito, donde el bucle repite cuando la condición “si” es verdadero y se detiene cuando es falso.

Figura 70

Bucle While True, Micropython

```
150 while True:
151     if modo_uno == 'K' and modo_dos == 'L': #Led rojo encendido y led azul apagado
152         first = carril.value() #llamar a carril con valor 1 o 0
153         modo_adulto() #Llamada de funcion
154         #print(modo_uno)
155         print('Modo Adulto')
156
157     if modo_dos == 'N' and modo_uno == 'L': #Led rojo apagado y led azul encendido
158         first = carril.value()
159         modo_infante()
160         #print(modo_dos)
161         print('Modo Infante')
162     if (modo_dos == 'L' and modo_uno == 'L') or (modo_dos == 'N' and modo_uno == 'N'):
163         sleep(1)
164         #print(modo_dos)
```

Nota. Elaboración propia.

- e. Dentro del bucle principal, la condición *If* se ejecuta siempre que la combinación de modo1 sea igual a un *string* “K” y el modo2 igual a “L”, como se muestra en la figura 71, dando paso al modo adulto.
- f. A partir del punto anterior, se realiza la petición de la función de modo adulto, dentro del modo adulto existe 2 opciones para el funcionamiento del motor, bien retrocede hacia la dirección del sensor de final de carrera o avanza un intervalo de posición como se muestra en la fila 102 del Anexo F1 y figura 71.

Figura 71
Función modo adulto, Micropython

```

102 def modo_adulto():
103     if first==0:
104         LED.on()
105         dirPin.high()
106         for x in range (1500):
107             #print('Izquierda')
108             stepPin.low()
109             time.sleep_us(600)
110             stepPin.high()
111             time.sleep_us(600)
112             if modo_uno == 'L':
113                 break
114     else:
115         LED.off()
116         # print('Derecha')
117         dirPin.low()
118         stepPin.low()
119         time.sleep_us(600)
120         stepPin.high()
121         time.sleep_us(600)
122

```

Nota. Elaboración propia.

- g. Dentro del bucle principal, la condición *If* se ejecuta siempre que la combinación de modo1 sea igual a un *string* “N” y el modo2 igual a “L”, como se muestra en la figura 69, dando paso al modo infante.
- h. Se realiza la petición de la función de modo infante, dentro del modo infante existe 2 opciones para el funcionamiento del motor paso a paso, bien retrocede hacia la dirección del sensor de final de carrera o avanza un intervalo de posición como se muestra en la fila 129 del Anexo F1 y figura 72.

Figura 72

Función modo infante, Micropython

```
125 def modo_infante():
126     if first==0:
127         LED.on()
128         dirPin.high()
129         for x in range (1300): # Rango de alcance
130             #print('Izquierda')
131             stepPin.low()
132             time.sleep_us(600) #Induccion de corriente para la Velocidad
133             stepPin.high()
134             time.sleep_us(600) #Velocidad
135             if modo_dos == 'L':
136                 #modo_dos = 'L'
```

Nota. Elaboración propia.

- i. Esta última opción de la condición *If* se ejecuta si siempre este es igual a modo1 y modo 2 a *string* “L”. Donde se manda a invernar al motor paso a paso como se muestra en la figura 72.

Modo de operación Infante. El modo de operación infante depende de la función (def modo_infante ():”), se tiene en cuenta la condición del desplazamiento y la velocidad del motor paso a paso, el cual está controlado por el controlador de motor paso a paso TB6600. Este controlador permite controlar la cantidad de pasos que el motor dará y la velocidad a la que se moverá.

En la Figura 66 está señalizado con una burbuja de letra f, y este modo se creó a partir del sensor de final de carrera, es decir cuando el sensor esta normalmente abierto, el motor empieza a girar en forma horaria y cuando el sensor esta normalmente cerrado este gira de forma antihoraria (dirPin.high()) un numero de pasos (for x in range (1300):) a razón de pasos que determinan la velocidad (time.sleep_us(600)) .

Modo de operación adulto. El modo de operación infante depende de la función (def modo_adulto ():”), se tiene en cuenta la condición del desplazamiento y la velocidad del motor paso a paso, el cual está controlado por el driver TB6600. Este controlador permite controlar la cantidad de pasos que el motor dará y la velocidad a la que se moverá.

A través de señales de pulso y dirección, el controlador envía los comandos necesarios para que el motor se mueva en la dirección deseada y a la velocidad especificada

En la Figura 66 está señalizado con una burbuja de letra h, y este modo se creó a partir del sensor de final de carrera, es decir cuando el sensor esta normalmente abierto, el motor empieza a girar en forma horaria y cuando el sensor esta normalmente cerrado esta gira de forma anti horaria (`dirPin.high()`) un numero de pasos (`for x in range (1500):`) a razón de pasos que determinan la velocidad (`time.sleep_us(600)`).

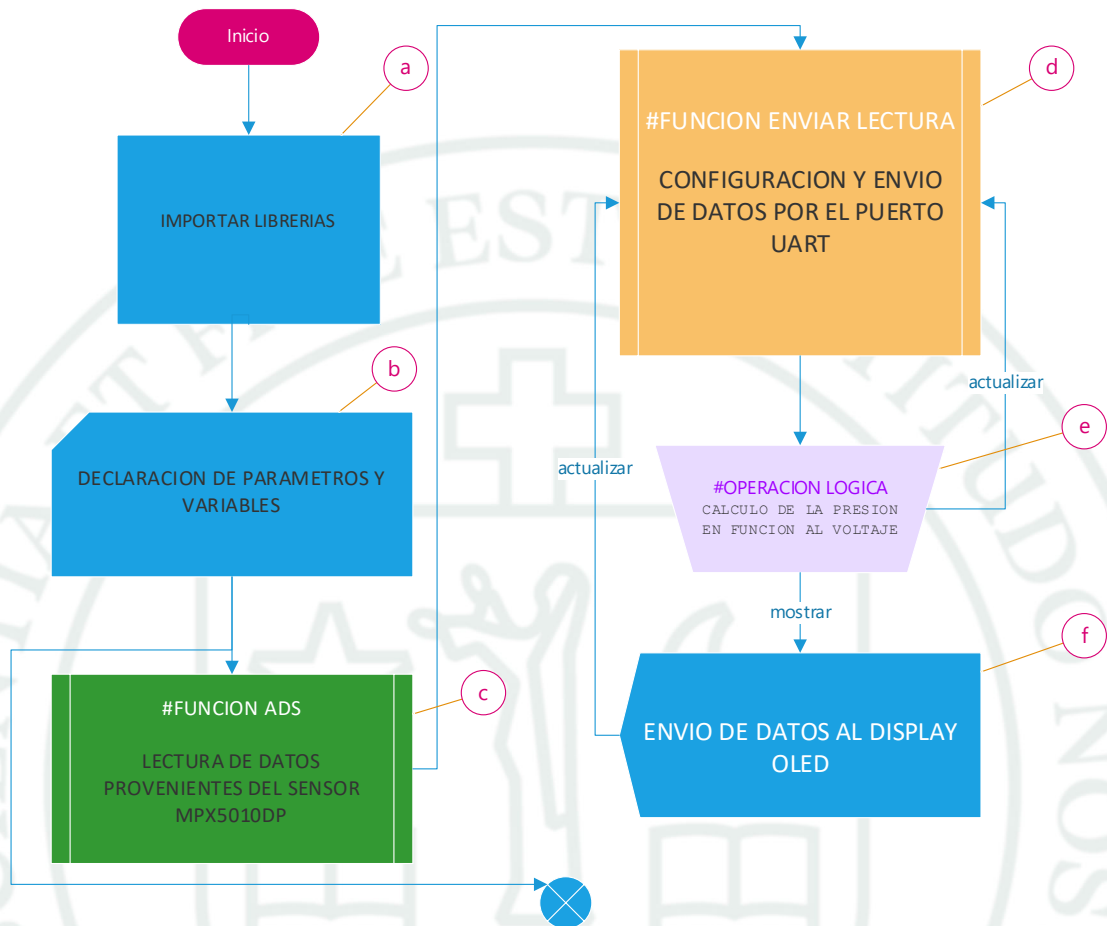
4.4.2 Control Remoto

A continuación, se muestra Figura 73 y Figura 74 que pertenecen al esquemático de flujo para la implementación del código de control manual, con Micropython, como se puede ver en el Anexo F2: Micropython R2.

En la Figura 73 se detalla el proceso para el control remoto parte 1. Comienza con la importación de librerías y paquetes predeterminados por el entorno de programación IDE Thonny. A continuación, se procede a la declaración de variables y parámetros necesarios para el funcionamiento del ventilador mecánico. Posteriormente, se definen funciones independientes, como el ADS y la función de envío de lecturas. Estas funciones operan lógicamente al leer el sensor, actualizar la información en intervalos de tiempo específicos y mostrar los datos en el display.

Figura 73

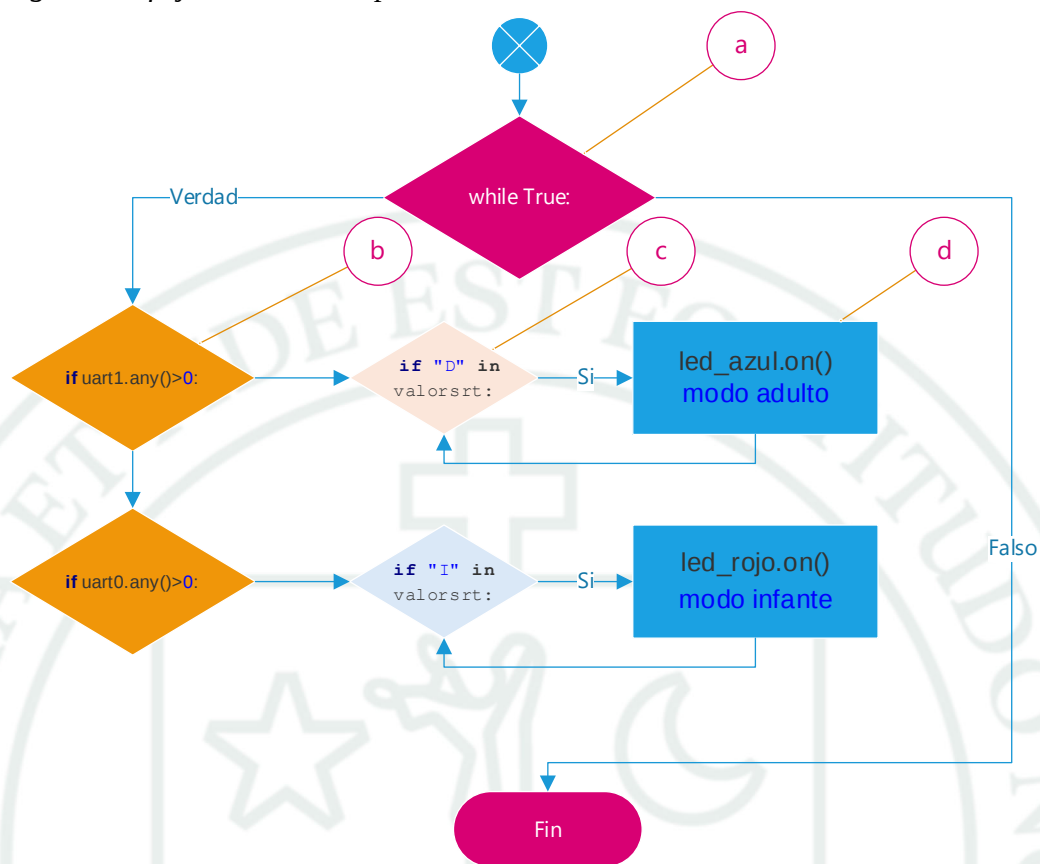
Diagrama de flujo, modo remote parte 1



Nota. Elaboración propia.

En la Figura 74 se representa la segunda parte del proceso de control remoto, comienza con un bucle infinito que constantemente lee los puertos. Si se detecta el carácter 'T' o 'D' en los puertos, se activan los leds, los cuales transmiten estos cambios de estado hacia el primer microcontrolador en la parte local y si no existe nada el proceso termina.

Figura 74
Diagrama de flujo, modo remoto parte 2



Nota. Elaboración propia.

Programación

La programación del microcontrolador para el control remoto del ventilador mecánico del Anexo F1: Micropython R2, se basa en la secuencia lógica del Figura 73 y Figura 74, que exhibe una serie de burbujas numeradas alfabéticamente desde la 'a' hasta la 'j'. A continuación, se describe:

- a. Inicialmente, se importaron las librerías necesarias para utilizar los módulos electrónicos, realizar operaciones matemáticas y manejar los puertos de comunicación. Estos detalles se encuentran expuestos con mayor profundidad en el Anexo F2: MicroPython R2, especialmente entre las filas 2 y 12. Dentro de estos módulos se incluyen el SSD1306, I2C, MATCH, UART, entre otros.

Figura 75

Importación de librerías y funciones, Micropython

```
1  #LIBRERIAS
2  from machine import UART, Pin
3  from machine import Timer
4  from machine import ADC
5  import time
6  import machine
7  from math import sqrt
8  from machine import Pin, SPI
9  from ssd1306 import SSD1306_SPI
10 from time import sleep
11 import utime
12 from machine import I2C, Pin
```

Nota. Elaboración propia.

- b. A continuación, se lleva a cabo la declaración de variables y parámetros en el código, abarcando desde la fila 15 hasta la 22 del Anexo F2. Este segmento se centra en la asignación de pines para el display OLED, así como en la configuración de la comunicación I2C.

Figura 76

Declaración de parámetros y puertos, Micropython

```
15 dev= I2C(1, scl=Pin(15), sda=Pin(14))
16 print(dev.scan())
17 address=72
18
19 ANCHO = 128
20 ALTO = 64
21 spi = SPI(0, 100000, mosi=Pin(19), sck=Pin(18))
22 oled = SSD1306_SPI(ANCHO, ALTO, spi, Pin(17),Pin(20), Pin(16))
```

Nota. Elaboración propia.

- c. Esta función se deriva de la librería ADS1X15, adaptándola a una resolución de ADC de 16 bits y a la comunicación I2C característica del módulo ADS1115. Esta opción resulta óptima para lograr una mayor precisión en la lectura de datos provenientes de sensores analógicos. Es importante recordar que la Raspberry Pi Pico tiene una resolución de 12 bits y admite voltajes inferiores a los 3.3 voltios. Por otro lado, el sensor MPX5010DP opera con 5 voltios, tal como se detalla en las filas 35 a 58 de la Figura 77.

Figura 77

Función ADS1X15, Micropython

```
34 #FUNCIONES DE LECTURA Y CONVERSION PARA EL MODULO ADS1115
35 def readConfig():
36     dev.writeto(address, bytearray([1]))
37     result = dev.readfrom(address,2)
38     return result[0]<<8 | result[0]
39 def readValueFrom(channel):
40     config = readConfig()
41     config &= ~(7<<12) #clear MUX bits
42     config &= ~(7<<9) #clear PGA
43     config |= (7 & (4+channel))<<12
44     config |= (1<<15)#trigger next convert
45     config |= (1<<9)#gain 4.096 volts
46     config = [int(config>>i & 0xff) for i in[8,0]]
47     dev.writeto(address, bytearray ([1] + config))
48     config = readConfig()
49     while (config & 0x8000) == 0:
50         pass
```

Nota. Elaboración propia.

- d. Esta función se encarga de leer el puerto I2C asociado al módulo ADS1115.

Luego, transmite los datos a través de los puertos de comunicación UART 0 y UART 1 utilizando un temporizador con un periodo de 50 ms, tal como se detalla en la fila 121 y fila 123 del Anexo F2: MicroPython R2 y en la Figura 78.

Figura 78

Función de lectura I2C, Micropython

```
121 uart1= UART(1,baudrate=9600, tx=Pin(4), rx=Pin(5))
122 uart0= UART(0,baudrate=9600, tx=Pin(0), rx=Pin(1))
123 tim = Timer(period=50, mode=Timer.PERIODIC, callback=EnviarLectura)
```

Nota. Elaboración propia.

- e. La operación lógica, considera los cálculos matemáticos respecto al sensor MPX5010DP, en la línea 71 del código se desarrolla la fórmula de para obtener la presión diferencial, a partir de la tensión de salida de la hoja de datos del fabricante, Anexo B2: Sensor MPX5010DP. También, se le agrega la ecuación de volumen obtenida del capítulo 3, Dimensionamiento del tubo de Venturi.

Figura 79

Operación de presión, Micropython

```
70 #Presion=((Voltaje-0.04*5-0.01)/(0.09*5))*100#Formula de DataSheet MPX5010DP KPa
71 Presion=((Voltaje-0.04*3.3+0.04)/(0.09*3.3))*10.1978#Formula de DataSheet MPX5010
72 # Presion= Vout-0.04*Vs +- Tol / 0.09Vs
73 # Vs= Voltaje de alimentación
```

Nota. Elaboración propia.

- f. Para concluir esta primera etapa del diagrama de flujo, se invoca al módulo de visualización OLED utilizando la comunicación SPI para presentar la lectura de presión del sensor. Este proceso se encuentra detallado en el código del Anexo F2, desde la fila 25 hasta la 30.
- g. En la segunda parte del diagrama de flujo, presentada en la Figura 74, se establece la condición "while True", característica de un bucle infinito común en la programación para microcontroladores en Python. Esto equivale, en cierta medida, al "void loop" utilizado en Arduino. Este segmento abarca desde la fila 128 hasta la 163 en el Anexo F2.
- h. La condición "if" de la línea 130 en el Anexo F2 se activa únicamente cuando hay datos presentes en el puerto serial UART 1, vinculado al dispositivo Bluetooth CH-06. Por otro lado, la condición "if" de la línea 147 se activa cuando hay datos en el puerto serial UART 0, asociado al dispositivo USB Serial.

Figura 80

Lectura de puertos parte 1, Micropython

```
130     if uart1.any()>0:
131         print ('detectado')
132         valor=uart1.read()
133         valorsrt=str(valor.decode('UTF-8'))
134
135         print (valorsrt)
136
137         if (valorsrt=='D'):
138             led_azul.on()
139             led_azul.off()
140             print('modo adulto')
141         elif (valorsrt=='I'):
142             led_rojo.on()
143             led_rojo.off()
144             print ('modo infante')
```

Nota. Elaboración propia.

- i. Luego, esta condición "if" se activa solo si el dato es del tipo string (D) o (I), tal como se especifica en el Anexo F2 y se muestra en la Figura 81.

Figura 81

Lectura de puertos parte 2, Micropython

```
130     #mueve el motor paso a paso de acuerdo al comando
131     if (valorsrt=='D'):
132         led_azul.on()
133         led_azul.off()
134         print('modo adulto')
135     elif (valorsrt=='I'):
136         led_rojo.on()
137         led_rojo.off()
138         print ('modo infante')
```

Nota. Elaboración propia.

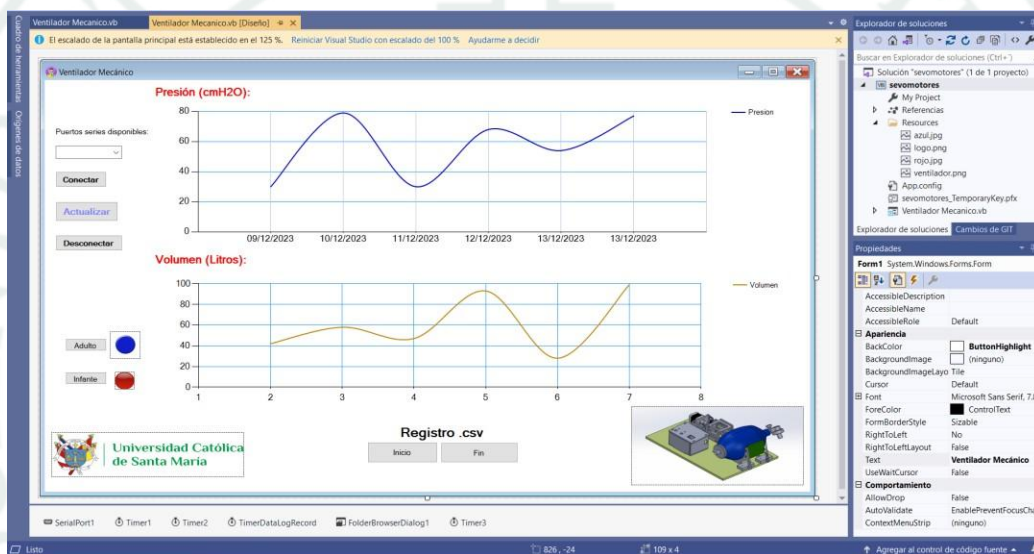
- j. Durante esta etapa, se activa el encendido del LED azul o rojo, generando así pulsos que alteran el estado de los pines de salida, como se declara en las líneas 125 y 126 del Anexo F2: MicroPython R2.

4.5 Desarrollo y Programación de GUI en Visual Basic

Para crear esta interfaz gráfica de usuario (GUI), se empleó el software Visual Basic 19.2 dentro del entorno de Aplicación de Windows Forms (.NET Framework). Se procedió inicialmente con el desarrollo del diseño de la interfaz gráfica, presentado en la Figura 82, mientras se configuraba simultáneamente cada función en su respectivo código principal, tal como se detalla en el Anexo F3: Visual Basic (GUI).

En la figura 82, se observa el diseño una ventana para seleccionar el puerto de entrada serial, además de botones destinados a las operaciones de actualización, conexión y desconexión. También se incluyeron dos botones para seleccionar los modos de operación, así como dos tablas que muestran las mediciones de presión y volumen de aire en función del tiempo. Adicionalmente, se implementó una base de datos para registrar las mediciones en un archivo .csv.

Figura 82
Gráfico de usuario del prototipo



Nota. Elaboración propia.

Programación

A continuación, se elabora el código de programación basado en los mockups, es decir, siguiendo la secuencia de los elementos diseñados, comenzando con:

- El botón "Conectar" responde desde la línea 28 hasta la 48 en el código del Anexo F3: Visual Basic (GUI). En la línea 31, se valida la conexión con puertos seriales a una velocidad de 9600 baudios. Posteriormente, se abre el puerto y, si la acción es exitosa, se despliega un cuadro de mensaje que indica "Conectando...", tal como se evidencia en la línea 83. Una vez establecida la

conexión, se inicia la lectura del puerto cada 10 ms utilizando un timer independiente.

Figura 83

Función conectar, Visual Basic

```

28 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
29     SerialPort1.PortName = ComboBox1.SelectedItem.ToString 'Obtener nombre de puerto selecc
30     'Label2.Text = ComboBox1.SelectedItem.ToString
31     SerialPort1.BaudRate = 9600 '9600 Baudios de velocidad
32     If SerialPort1.IsOpen Then
33         MessageBox.Show("Puerto Abierto, cierre el puerto y vuelva a intentarlo")
34     Else
35         Try
36             SerialPort1.Open() 'Abrir puerto serial
37             MessageBox.Show("Conectado...") 'Mostrar mensaje
38             Timer1.Interval = 10 'Ajustar Timer a intervalo de 1 segundo, el timer leera los
39             AddHandler Timer1.Tick, AddressOf ActualizarGrafica 'Generar Handler, cada vez c

```

Nota. Elaboración propia.

- La subfunción "actualizar" representada en la figura 84, abarca desde la línea 50 hasta la línea 97 en el código. Su propósito es leer el puerto serial, verificando la presencia de bytes disponibles en el buffer. Si hay datos, se decodifican para obtener el valor de la presión del aire en el prototipo del ventilador mecánico. Luego, se realizan cálculos y ajustes para determinar el volumen, siguiendo la ecuación detallada en el capítulo 3 sobre Dimensionamiento.

Figura 84

Función actualizar, Visual Basic

```

50 Public Sub ActualizarGrafica(sender As Object, e As EventArgs)
51
52     If SerialPort1.IsOpen Then
53         If SerialPort1.BytesToRead > 0 Then 'Si hay Bytes disponibles en el Buffer
54             LecturaSerial = SerialPort1.ReadLine 'Leer hasta que hay un terminador de la cac
55
56             Label2.Text = "P: " + LecturaSerial + " cmH2O" 'Asignar a etiqueta 2 el valor le
57
58             Dim Presion = LecturaSerial.Length
59             Dim PcmH2O = Presion / 10.1978
60             Dim V = ((2 * PcmH2O * 1000) / (1.225 * 76.96)) ^ (1 / 2)
61
62
63
64             Dim Area = 0.3397 * 10 ^ (-3)
65             Dim Qlitros = Area * V * 1000 * 60
66
67             Dim Qabsoluto = (Qlitros - 55)
68
69             Dim Vabsoluto = Math.Round((Qabsoluto / 60), 5) * 2
70             Dim resultado As String = Vabsoluto
71             Vol = resultado
72             Label3.Text = "V: " + resultado + " Litros"
73
74
75             Chart1.Series("Presion").Points.AddXY(DateTime.Now.ToLongTimeString, LecturaSerial)
76             Chart2.Series("Volumen").Points.AddXY(DateTime.Now.ToLongTimeString, resultado)

```

Nota. Elaboración propia.

- La línea 75 y 76 de la Figura 84, cumple la función de actualizar a las 2 graficas de presión y volumen de aire.
- Desde la línea 141 hasta la 148 en la Figura 85, se establece la programación para dimensionar a una escala los gráficos que representan la evolución temporal de los valores de presión y volumen generados por la ventilación.

Figura 85

Escalar los datos, Visual Basic

```

141 Chart1.ChartAreas(0).AxisY.Maximum = 30
142 Chart1.ChartAreas(0).AxisY.Minimum = -5
143 Chart1.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled = True
144 '-----Labeling values of the graph-----'
145 Chart2.ChartAreas(0).AxisY.Maximum = 1
146 Chart2.ChartAreas(0).AxisY.Minimum = -0.2
147 Chart2.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled = True
148 '-----Labeling values of the graph-----'
149 End Sub
150
151

```

Nota. Elaboración propia.

- Otra tarea relevante son los botones de pulsación mostrados en la Figura 86, las cuales abarcan desde la línea 111 hasta la 121 en el código. Ambos botones escriben en el puerto serial los strings "D" e "I".

Figura 86

Función botones, Visual Basic

```

111 Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
112     'Comando para adulto
113     SerialPort1.Write("D")
114     PictureBox2.Visible = True
115 End Sub
116
117 Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
118     'Comando para Infante
119     SerialPort1.Write("I")
120     PictureBox3.Visible = True
121 End Sub
122

```

Nota. Elaboración propia.

- El interfaz del prototipo incluye una opción para almacenar los datos de presión, flujo y la hora de cada muestra, tal como se muestra en la Figura 87. Estos registros se almacenan en una carpeta específica designada por el

usuario. El temporizador se activa a través de un botón y se detiene el registro de datos con otro botón adicional. Todos los datos se guardan en un archivo .csv para facilitar su análisis y procesamiento futuro.

Figura 87

Función de almacenamiento, Visual Basic

```
183 Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
184     Dim nombreArchivo As String = "datos" & contador.ToString() & ".csv"
185     archivoCSV = New StreamWriter("D:\Lectura\" & nombreArchivo)
186     Timer3.Interval = 10
187     Timer3.Start()
188 End Sub
189
190 Private Sub Timer3_Tick(sender As Object, e As EventArgs) Handles Timer3.Tick
191     Dim tiempo As String = DateTime.Now.ToString("HH:mm:ss")
192     Dim datos As String = LecturaSerial
193     Dim datos1 As String = Vol
194     Dim datosCompleto As String = tiempo & "," & datos & "," & datos1
195     archivoCSV.WriteLine(datosCompleto)
196     archivoCSV.Flush()
197     contador += 1
198 End Sub
199
200 Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
201     Timer3.Stop()
202     archivoCSV.Close()
203 End Sub
204
```

Nota. Elaboración propia.

Finalmente, el interfaz gráfico de usuario (GUI) ofrece diversas opciones. En el panel izquierdo, se encuentra la posibilidad de seleccionar el puerto serial y establecer la conexión correspondiente. Asimismo, presenta dos gráficos que exhiben la señal de presión y el caudal del ventilador mecánico. En la parte superior del gráfico se muestra la presión en cmH₂O, mientras que en la parte inferior se visualiza el caudal en litros por segundo.

El GUI también integra el control de los modos de operación, tanto para infantes como para adultos. El modo infante se indica mediante un LED rojo, mientras que el modo adulto se señala con un LED azul. Además, el GUI incorpora una función de registro de muestras de manera independiente. Esta función se activa mediante un botón de inicio y se detiene con otro botón. Todos los datos registrados se almacenan en un archivo .csv para un análisis posterior.

4.6 Desarrollo de la APP

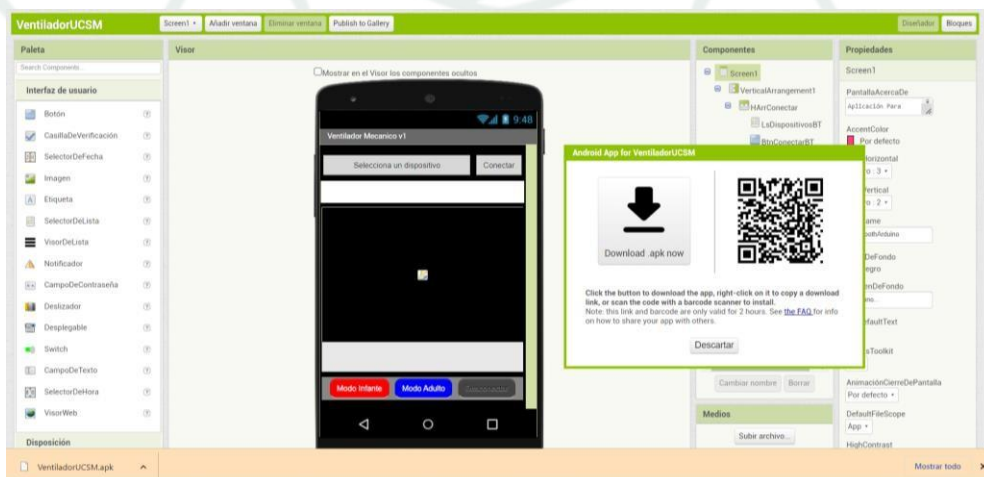
La aplicación para teléfonos móviles tiene la función de mostrar la gráfica de la señal de presión y también permite cambiar los modos de operación del ventilador mecánico. Esta aplicación se desarrolló utilizando el software virtual de App Inventor, una herramienta creada en colaboración con el Instituto Tecnológico de Massachusetts (MIT). Utiliza un lenguaje de programación basado en bloques, que se asemeja a piezas de un juego de construcción.

En primer lugar, se diseñó la interfaz gráfica como se exhibe en la Figura 88. Esta interfaz incluye botones para la conexión serial inalámbrica a través de Bluetooth con el ventilador mecánico. Esta conexión permite la recepción de datos desde el sensor, así como el envío de los modos de operación del ventilador mecánico, tanto para el modo adulto como para el modo infante.

Posteriormente, basándose en la descripción anterior, se elaboró la secuencia de código utilizando bloques de colores y formas, cada uno con una tarea y función específica. Esta secuencia de bloques se detalla en el Anexo F4: Aplicación en APP Inventor.

Figura 88

Interfaz gráfica de usuario para teléfono móvil

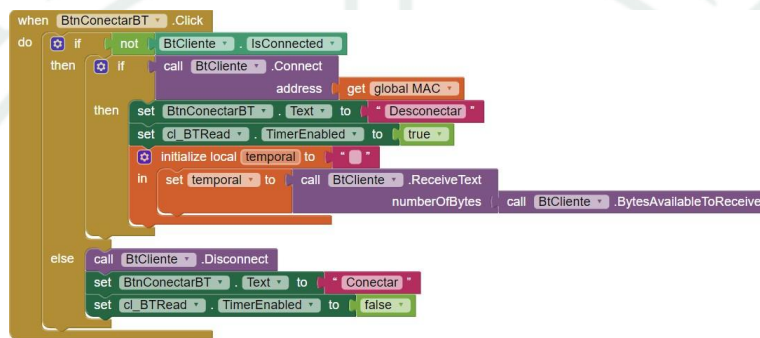


Nota. Elaboración propia.

Programación

- En la Figura 89 se muestra la conexión entre el prototipo y un teléfono móvil a través de Bluetooth. En esta imagen, se visualiza la lectura de todos los dispositivos disponibles, seguida por la elección y selección de un dispositivo específico.

Figura 89
Conexión mediante bluetooth, APP Inventor

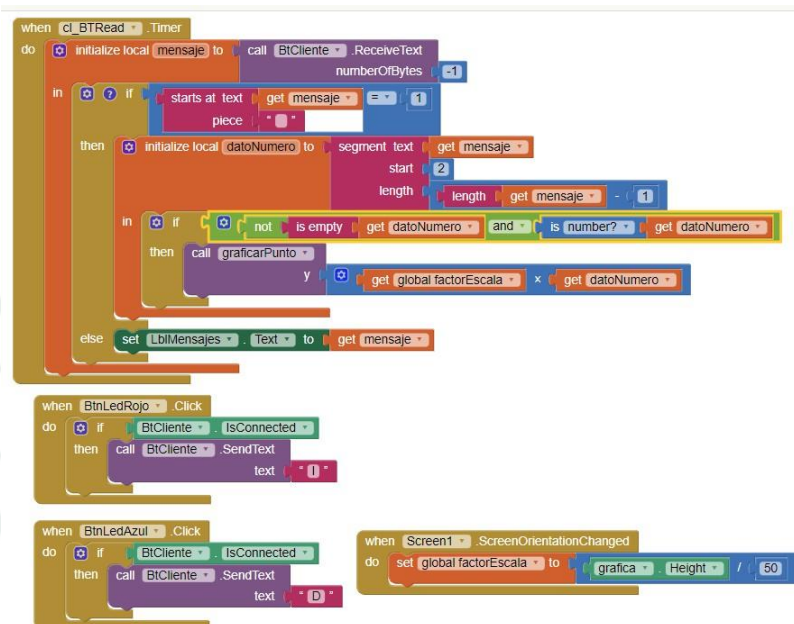


Nota. Elaboración propia.

- Luego en la figura 90, el bucle "cl_BTRead", sincronizado con un temporizador, se encarga de leer y descifrar el mensaje, mientras que el bucle "graficarPunto" se encarga de representar gráficamente la señal del sensor de presión. Finalmente, se presentan las dos funciones para seleccionar y deseleccionar el modo de operación del prototipo del ventilador mecánico.

Figura 90

Lectura y escritura de datos seriales, APP Inventor



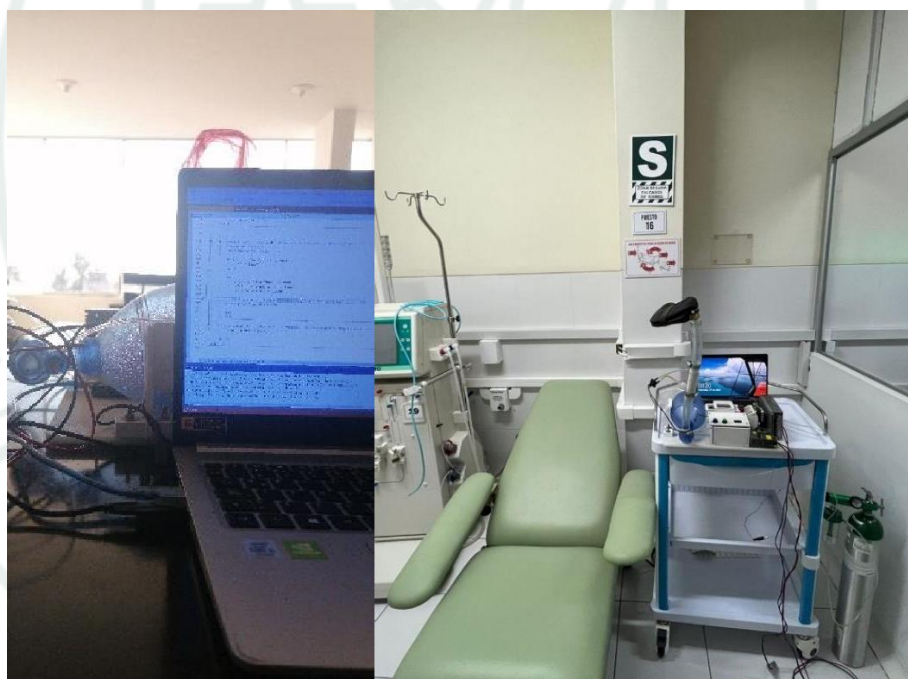
Nota. Elaboración propia.

CAPITULO V

PRUEBAS Y RESULTADOS

En este capítulo se lleva a cabo la calibración y las pruebas de funcionamiento del prototipo en sus dos modos, así como el análisis de dichas pruebas en términos de la fiabilidad del prototipo. Para ello, se realizaron las pruebas en un centro médico ubicado en la ciudad de Arequipa, un entorno con una ventilación óptima, niveles bajos de ruido y una iluminación adecuada, como se puede apreciar en la figura 91. El ventilador se conectó a una fuente de alimentación de 220 voltios. Además, se empleó un ordenador para recopilar los datos y un teléfono móvil para controlar el prototipo mediante bluetooth y medir los niveles de ruido generados por el ventilador mecánico.

Figura 91
Fotografías del prototipo



Nota. Elaboración propia.

5.1 Calibración del prototipo

El proceso de calibración es esencial para asegurar el óptimo funcionamiento de cualquier dispositivo electrónico. En el caso del prototipo de ventilador mecánico, operando en sus dos modos, la calibración precisa resulta crucial. Esta tarea implica ajustar tanto la velocidad como el torque del motor paso a paso, aspectos que están directamente vinculados a la configuración del driver TB6600 y al código programado en Python, especialmente en la función de bucle "for". De esta precisión en la calibración depende tanto la presión generada por la bolsa de Ambu como el nivel de ruido producido por el motor, como se muestra en la Figura 92.

Figura 92

Configuración del driver y del código



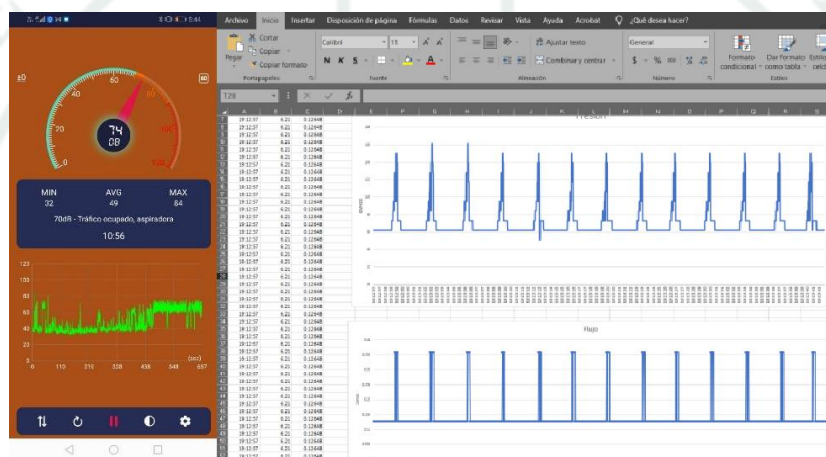
Nota. Elaboración propia.

La configuración tanto del driver como del código genera diferentes intervalos de pico de presión máxima y/o volumen, así como distintos niveles de ruido. Este último aspecto es particularmente relevante y se abordó mediante la medición del nivel de ruido. Para llevar a cabo esta tarea, se empleó una aplicación móvil llamada "Sonómetro", disponible en Play Store, como se muestra en la figura 93 y en el Anexo G3. Esta aplicación permitió la captura precisa de los niveles de ruido, un factor recomendado por

expertos y mencionado en la tabla 2 del Capítulo 3 en términos de consideraciones de fabricación.

Además, las mediciones recopiladas se almacenaron a través de la interfaz del usuario, lo que permitió guardar los datos en formato de archivo .csv. Estos datos se representaron gráficamente utilizando Excel, como se muestra en el lado derecho de la Figura 93.

Figura 93
Sonómetro y generación de gráficos con Excel



Nota. Elaboración propia.

5.1.1 Testeo y adquisición de data

El proceso de testeo implica someter el dispositivo a diferentes intervalos para adquirir datos sobre los tiempos de los picos más altos de la presión y el nivel de ruido generado por el motor paso a paso. Estos datos se obtienen al modificar la velocidad y el torque del driver TB6600, así como ajustar el rango de alcance de la varilla sinfín en el bucle "for" del código.

La Tabla 21 presenta los resultados del testeo derivados de la configuración. En la primera columna se asigna la corriente óptima necesaria para el funcionamiento del

motor NEMA 17 junto con su respectivo paso. Estos valores de paso y rango se insertan en el código local del Anexo F1: Micropython R1.

El funcionamiento del ventilador produce diferentes niveles de ruido según la configuración recibida. Para medir el ruido, se empleó la aplicación móvil "Sonómetro" disponible en Play Store, visible en la cuarta columna de la Tabla 20 y en el Anexo G3: Captura de pantalla de la tabla experimental.

Otro indicador relevante es el ciclo de repetición en segundos, derivado de la configuración del rango en el código. Este ciclo se relaciona con los picos altos de presión del ventilador mecánico, como se observa en la quinta columna de la Tabla 21 y en el Anexo G3: Captura de pantalla de la tabla experimental. Además, se realiza la conversión de estos ciclos de repetición a respiraciones por minuto (rpm).

Tabla 21*Datos generados a partir de los indicadores*

Configuración TB6600											
S1	S2	S3	S4	S5	S6	Ruido			Pre		
						Paso	Rango	(dB)	Segundos	(cmH2O)	rpm
ON	OFF	OFF	ON	OFF	ON	500	1500	75	3.5	5.67	17
							1700	74	4.2	6.77	14
							1600	74	4.5	7.87	13
							1800	70	5	8.97	12
							2000	70	5	7.87	12
OFF	ON	ON	ON	OFF	ON	400	1700	70	3.5	6.77	17
							1800	70	4	7.87	15
							1900	64	3.8	7.87	16
							2100	64	4	8.97	15
							2200	59	3.8	7.87	16
ON	OFF	ON	ON	OFF	ON	300	2200	64	2.5	5.67	24
							2400	67	2.6	6.77	23
							2500	64	3.6	7.87	18
							2700	64	4.2	7.87	14
							3000	57	4.5	8.97	13
ON	ON	OFF	ON	OFF	ON	200	3400	63	3.6	6.77	17
							3500	58	3.8	7.87	16
							3800	57	4.2	12.28	14
							4000	57	4	13.39	15
							4400	54	3.5	14.49	17

Nota. Elaboración propia.

5.1.2 Consideraciones Médicas

Para que el ventilador funcione en el modo infante o modo adulto, se consideran los estándares de respiración por minuto según la edad de las personas, tal como se muestra en la tabla 22. elaborada por el Dr. Alex Velázquez, muestra las respiraciones por minuto según la edad. Se tomó como referencia la fila correspondiente a edades entre 6 y 12 años para configurar el modo infante y la fila de 60 más para el modo adulto.

Tabla 22
Respiración por minuto según la edad

EDAD	rpm		
Neonato	30	50	40
0-5 meses	25	45	35
6-12 meses	20	40	30
1-3 años	20	35	27.5
3-5 años	20	30	25
6-12 años	15	30	22.5
13-18 años	14	20	17
18-60 años	12	20	16
60 mas	12	16	14

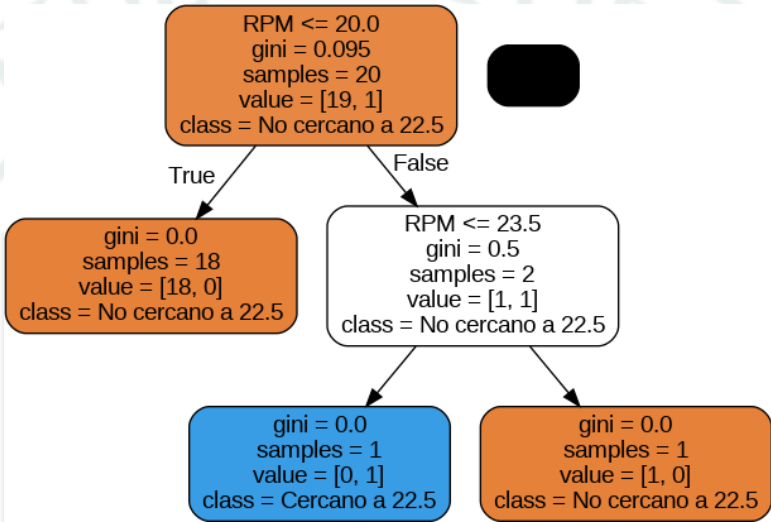
Nota. Tomado de Dr. Alex Velázquez.

5.1.3 Calibración modo infante

Para calibrar el modo infantil, se llevará a cabo un estudio de estadística descriptiva para analizar la interdependencia entre dos o más variables, como sugieren (Romero, Lázaro, & González). Esto implica seleccionar la configuración más adecuada o cercana a las revoluciones por minuto (rpm) del infante, así como minimizar el ruido

generado por el motor. Este proceso se realiza en Google Colab, donde se ingesta los datos de la Tabla 22. Utilizando la biblioteca Sklearn, específicamente el método de clasificación por árbol de decisión, se selecciona la mejor opción para 22.5 rpm como se muestra en la figura 94, más detalles del código en el Anexo J.

Figura 94
Método de árbol de clasificación del modo infante



Nota. Elaboración propia.

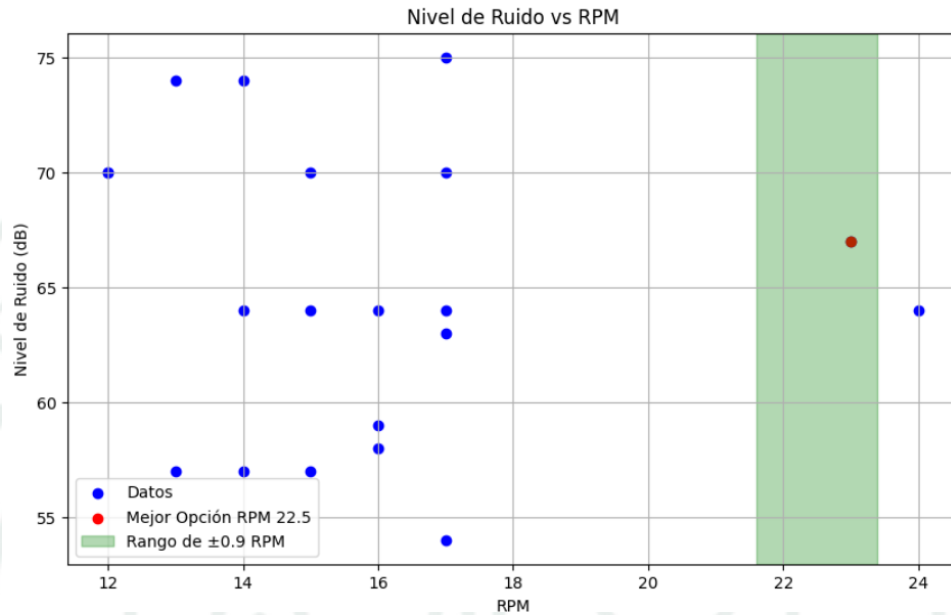
A continuación, se presenta un diagrama descriptivo que destaca la mejor opción para el funcionamiento en modo infantil, elaborado con Google Colab. Según la tabla 22, el estándar para la respiración por minuto es de 22.5 rpm. Por lo tanto, en la Figura 95 se representa la configuración que se aproxima a este estándar, mostrando en la franja los resultados con un margen de error estándar de +/- 0.9 rpm. Esta configuración se basa en un paso de 300 y un rango de 2400, con 23rpm y 67dB, más detalles en el Anexo J.

Figura 95

Distribución de ruido y rpm modo infante

La mejor opción para RPM cercano a 22.5 es:

Paso 300.00
Rango 2400.00
dB 67.00
Tiempo 2.60
Presión 6.77
rpm 23.00
Name: 11, dtype: float64



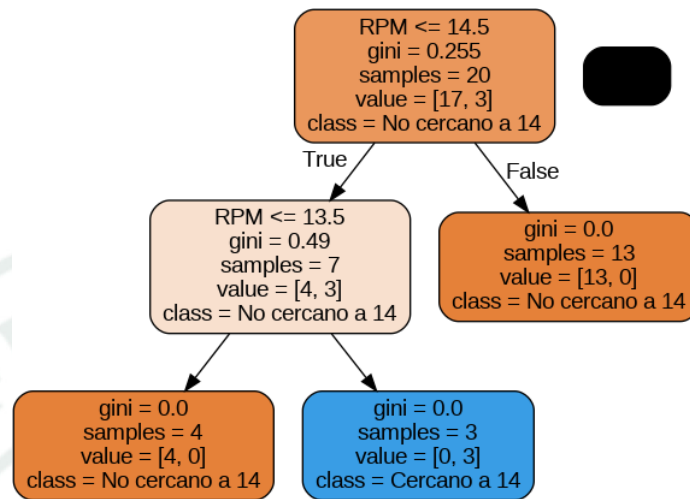
Nota. Elaboración propia.

5.1.4 Calibración modo adulto

Para calibrar el modo adulto, se llevará a cabo un estudio de estadística descriptiva para analizar la interdependencia entre dos o más variables, como sugieren (Romero, Lázaro, & González). Esto implica seleccionar la configuración más adecuada o cercana a las revoluciones por minuto (rpm) del adulto, así como minimizar el ruido generado por el motor. Este proceso se realiza en Google Colab, donde se ingresa los datos de la Tabla 22. Utilizando la biblioteca Sklearn, específicamente el método de clasificación por árbol de decisión, se selecciona la mejor opción para 14 rpm como se muestra en la figura 96, más detalles del código en el Anexo J.

Figura 96

Método de árbol de clasificación del modo adulto



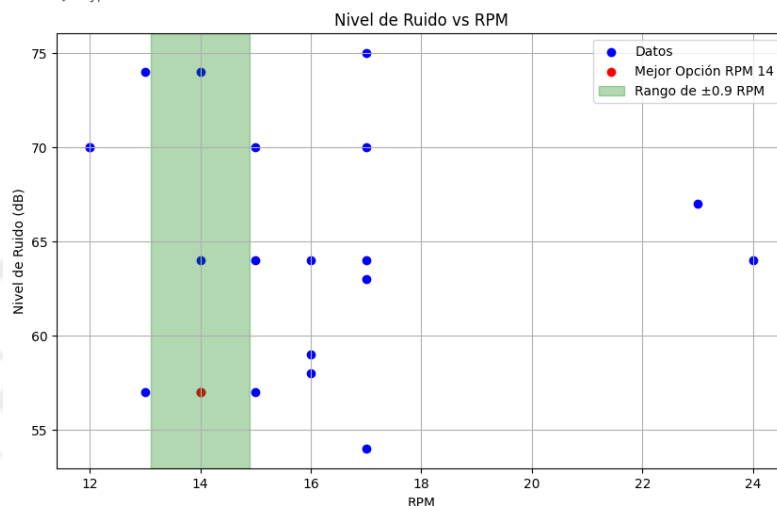
Nota. Elaboración propia.

A continuación, se presenta un diagrama descriptivo que destaca la mejor opción para el funcionamiento en modo infantil, elaborado con Google Colab. Según la tabla 22, el estándar para la respiración por minuto es de 14 rpm. Por lo tanto, en la Figura 97 se representa la configuración que se aproxima a este estándar, mostrando en la franja los resultados con un margen de error estándar de ± 0.9 rpm. Esta configuración se basa en un paso de 200 y un rango de 3800, con 14rpm y 57dB, más detalles en el Anexo J.

Figura 97

Distribución de ruido y rpm de modo adulto

La mejor opción para RPM cercano a 14 es:
Paso 200.00
Rango 3800.00
dB 57.00
Tiempo 4.20
Presión 12.28
rpm 14.00
Name: 17, dtype: float64



Nota. Elaboración propia.

5.2 Prueba de funcionamiento del prototipo

Teniendo en cuenta la calibración requerida para los modos adulto e infantil, con respiración por minuto de 23 RPM y 14 RPM respectivamente, se configuran los parámetros en el microcontrolador Raspberry Pi Pico en el bucle “for” con un paso de 300 y 200, con rangos de 2400 y 3800. El inicio de puesta en marcha del ventilador se lleva a cabo desde un dispositivo móvil Android a través de conexión Bluetooth, así como desde una computadora con sistema operativo Windows 10 mediante comunicación RS232. Además, se incluye un control de mando local, como se ilustra en la Figura 98.

Esta prueba tiene como objetivo facilitar la visualización de la presión ejercida por la bolsa de Ambu en el display del control local y en la computadora. Asimismo, los datos recopilados en la computadora serán registrados en archivos Excel para permitir una interpretación detallada de la confiabilidad del prototipo.

Figura 98
 Fotografías de las pruebas del prototipo



Nota. Elaboración propia.

Las configuraciones para los dos modos de operación se detallan en la Tabla 23, donde se especifica la configuración hardware asociada al TB6600 y el software correspondiente al código del microcontrolador.

Tabla 23
 Configuración hardware y software de modos de operación

	Configuración	Modo Infante	Modo Adulto
Hardware	TB6600	ON ON OFF ON OFF ON	ON ON OFF ON OFF ON
	for x in range ():	2400	3800
Software	time.sleep_us()	300	200

Nota. Elaboración propia.

5.2.1 Prueba de los controles de operación

El prototipo cuenta con tres modos de control: local, remoto alámbrico e inalámbrico. Durante estas pruebas, es crucial validar su funcionamiento, ya que cada uno de estos modos cumple un propósito específico.

Control local

El control manual o local se lleva a cabo a través de dos pulsadores. El botón azul se utiliza para seleccionar el modo de operación para adultos, mientras que el botón negro se emplea para activar el modo para infantes, tal como se muestra en la Figura 99. Además, se incluyen dos LEDs que indican el estado de operación, según se detalla en la Tabla 24.

Tabla 24
Estado de operación

Estado de operación	Led Azul	Led Rojo
Infante	OFF	ON
Adulto	ON	OFF
Neutro	ON	ON
Neutro	OFF	OFF

Nota. Elaboración propia.

El tiempo de respuesta para iniciar el modo de operación del prototipo varía entre 1 y 2 segundos, lo que implica un arranque no inmediato debido al uso de pulsadores con enclavamiento. Esta demora se debe a las interrupciones integradas en el código, que funcionan a través de contadores. En la pantalla, se muestra una presión de 5.16 cmH₂O, la cual se encuentra dentro del rango habitual de ventilación para ventiladores mecánicos (Bugedo, 2022), que oscila típicamente entre 4 y 6 cmH₂O.

Figura 99

Pulsadores de modo de operación local



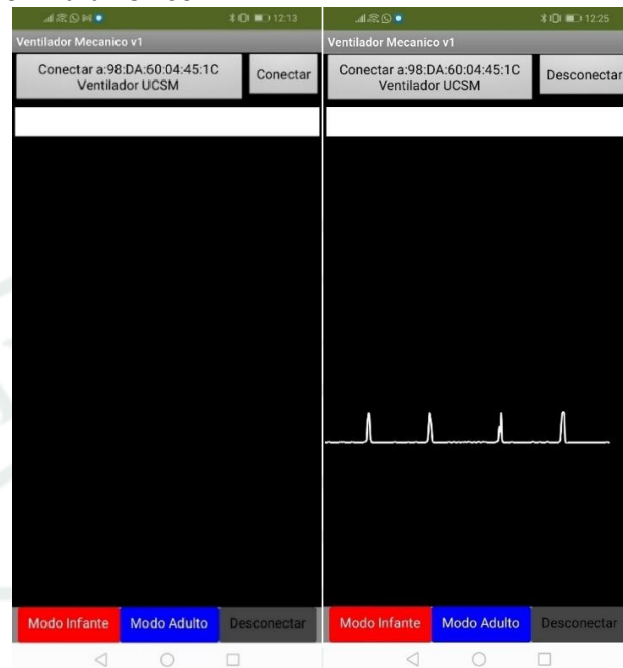
Nota. Elaboración propia.

Control Inalámbrico

El modo de control remoto inalámbrico se basa en el uso del módulo Bluetooth CH-06. El dispositivo se identifica como "Ventilador UCSM" y se accede mediante la contraseña "1234". Además, se ha desarrollado una aplicación móvil utilizando MIT App Inventor, como se muestra en la Figura 100. La comunicación se realiza a través de serie a 9600 baudios.

La respuesta de los modos de operación es prácticamente instantánea a distancias menores de 7 metros. Sin embargo, a distancias mayores, la latencia puede aumentar o puede que no se responda a los comandos del modo infante o adulto. Específicamente, a menos de 7 metros, la respuesta se registra en 500 milisegundos, mientras que, a distancias entre 7 y 16 metros, la latencia varía entre 600 y 1600 milisegundos. La pantalla muestra la señal de presión del ventilador mecánico y ofrece ambas opciones para la elección del modo de operación.

Figura 100
Pantalla de control Inalámbrico



Nota. Elaboración propia.

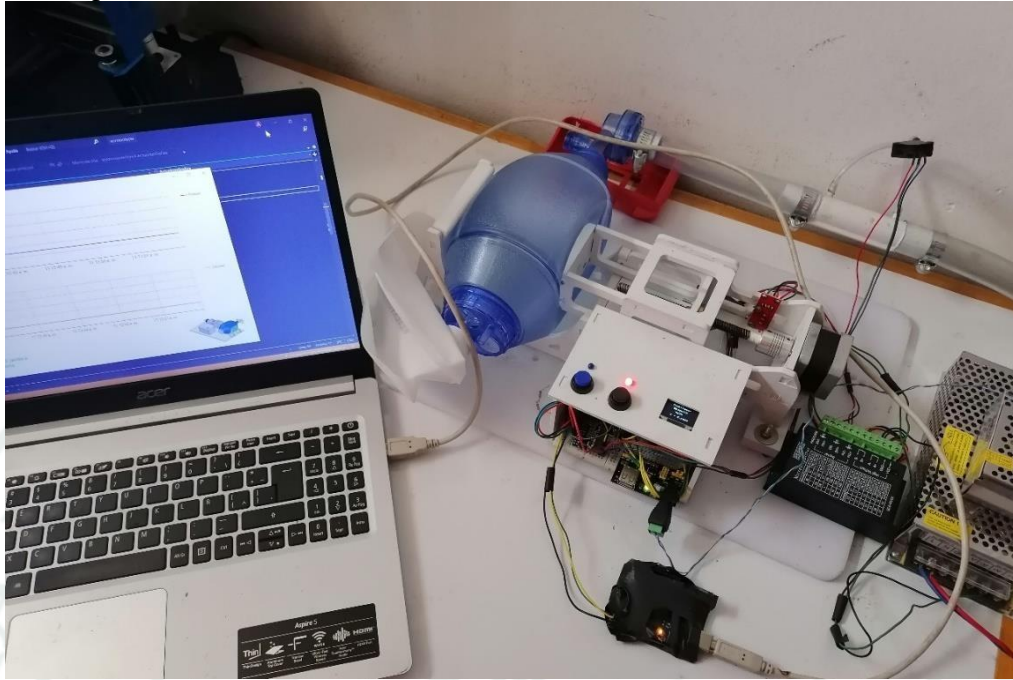
Control Alámbrico

El modo de control remoto alámbrico se basa en la comunicación serial por cable (RS232), utilizando el puerto USB serial a 9600 baudios. Se ha desarrollado una interfaz gráfica de usuario en Visual Basic, como se muestra en la Figura 101. Esta conexión tiene un alcance de hasta 7 metros y ofrece una respuesta casi instantánea.

La función principal de la interfaz gráfica de usuario (GUI) es monitorear la señal de presión y volumen, lo cual es crucial para configurar el prototipo y la adquisición de data, ya que la presión y el volumen ejercen una influencia significativa.

Figura 101

Control por comunicación serial USB



Nota. Elaboración propia.

5.2.2 Prueba de la presión de aire y volumen

Los gráficos generados en el GUI son resultado de la lectura del sensor de presión diferencial MPX5010DP, que capta las señales de presión del aire generado por la compresión de la bolsa Ambu. Este sensor utiliza la técnica del tubo de Venturi, donde la presión disminuye debido a la variación en la sección, lo que resulta en un incremento en la velocidad del fluido (Zabala Jorquera, 2014).

Presión de Aire

Gracias a la ecuación de la hoja de fabricantes del Anexo B2: Sensor MPX5010DP, es posible obtener la ecuación del valor de la presión, donde:

$$V_{out} = V_s \times (0.09 \times P + 0.04) \pm 5.0\%V_{FSS}$$

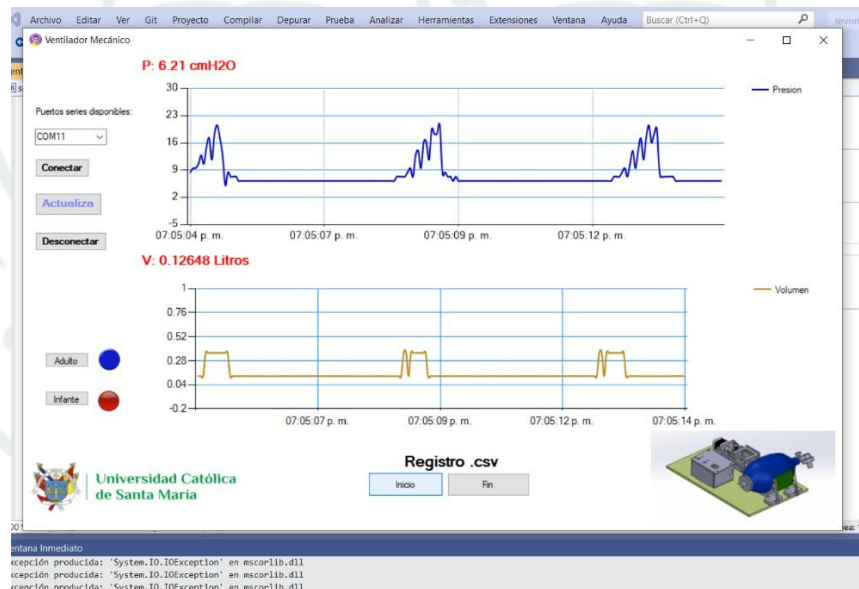
$$P = \frac{((V_{out} - 0.04 \times V_s \pm 0.04))}{0.09 \times V_s}$$

$$P = \frac{V_{out} - 0.04 \times 3.3 + 0.04}{0.09 \times 3.3} \text{ KPa}$$

$$P = \frac{V_{out} - 0.04 \times 3.3 + 0.04}{(0.09 \times 3.3)(10.1978)} \text{ cmH}_2\text{O}$$

La ecuación mencionada anteriormente se implementa en el código de Visual Basic, donde el sensor envía valores en milivoltios para el voltaje de salida (V_{out}) y 3.3 voltios es el valor pico (V_s). Con estas configuraciones detalladas, el ventilador produce picos de presión de aproximadamente 15 y 20 cmH₂O tanto en el modo de operación adulto como en el modo infante, como se muestra en la Figura 102 y la Figura 103. En ambos modos, la presión positiva al final de la espiración (PEEP) oscila entre 5 y 6 cmH₂O, cumpliendo así con los estándares médicos requeridos para un ventilador mecánico.

Figura 102
Respuesta del modo de operación adulto



Nota. Elaboración propia.

Volumen de Aire

El volumen de aire depende de la presión generada y también es posible estimar a partir de la ecuación obtenida en Dimensionamiento:

$$V_1 = \sqrt{\frac{2 * \Delta P}{\rho * 76.96}} \quad (16)$$

Donde:

- V=Velocidad m/s
- AP=Variación de presión en (KPa) y 1 kPa = 101.972 kg/m²
- P=densidad del aire 1.225 kg/m³

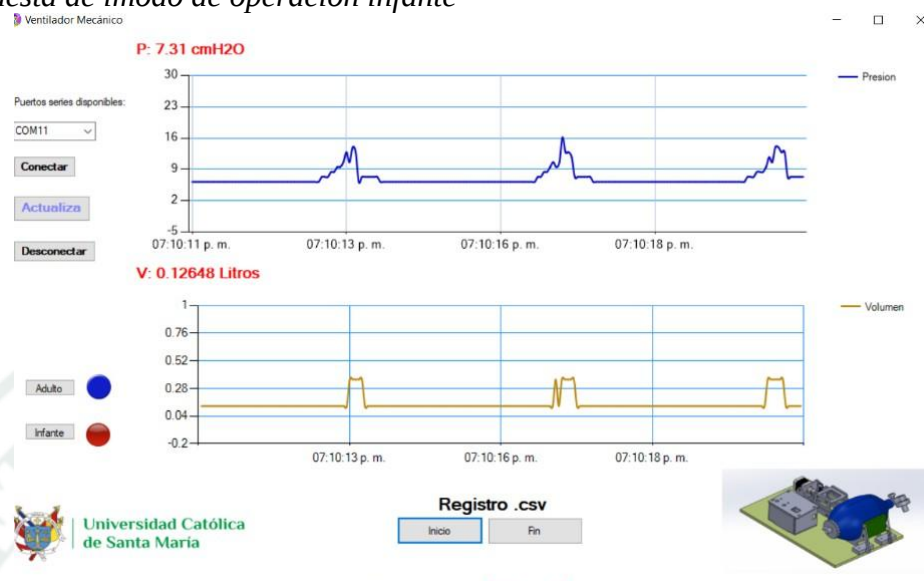
Además, en Caudal Q es igual:

$$Q = A_1 * V_1 = \text{constante} \quad (17)$$

Entonces, *Volumen = Area × Velocidad × tiempo*

Las ecuaciones anteriores se implementaron en los códigos de Visual Basic, generando volúmenes pico de 0.31 litros en el modo de operación infante y 0.42 litros en el modo de operación adulto, tal como se muestra en las Figuras 102 y 103.

Figura 103
Respuesta de lmodo de operación infante



Nota. Elaboración propia.

5.3 Resultado de presión y rpm

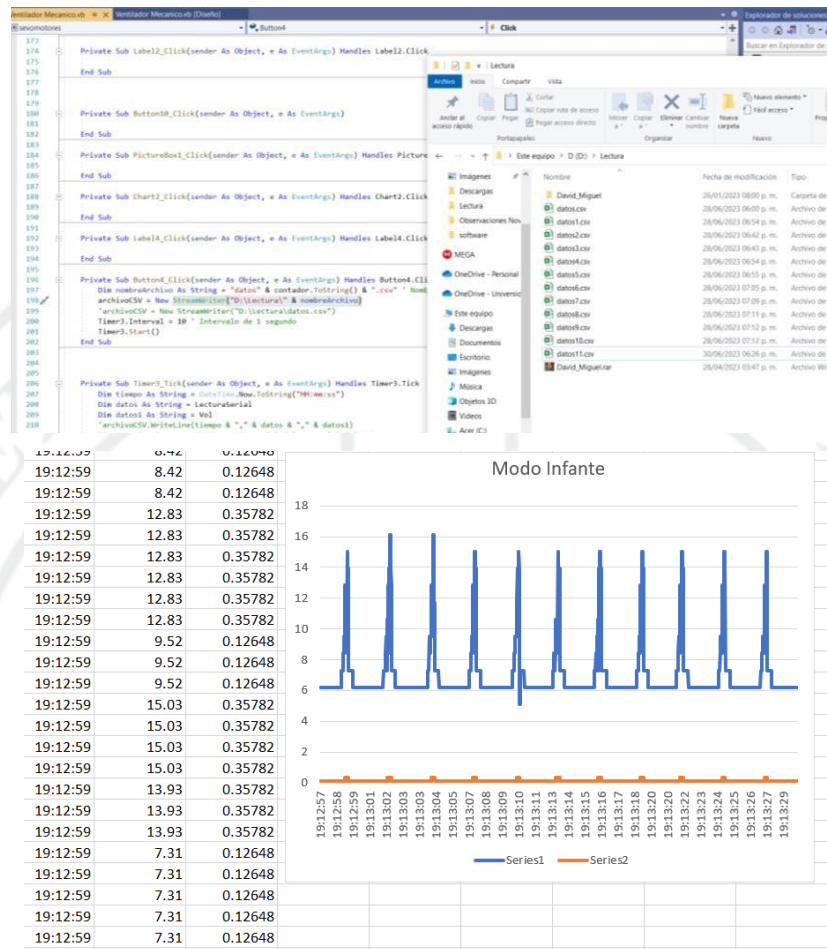
Uno de los criterios de los resultados de funcionamiento es la presión y la respiración por minuto (rpm) en ambos modos de operación.

5.3.1 Modo Infante

Para el modo infante se configuró el driver TB6600 en (ON OFF ON ON OFF ON), lo que se traduce en un rango de 2400 y 300 pasos en el código de MicroPython, detallado en la tabla 21. Los resultados de las señales de variación de la presión en el tiempo se obtienen al pulsar el botón "Registro .csv", como se muestra en la figura 104. Este genera archivos con extensión .csv y los almacena en una carpeta del computador con el nombre de "datos", tal como se muestra en la Figura 104. Los datos capturados previamente se pueden visualizar en Excel, con picos que se aproximan a los 16 cmH2O.

Figura 104

Captura de datos de rpm con Excel, modo infante

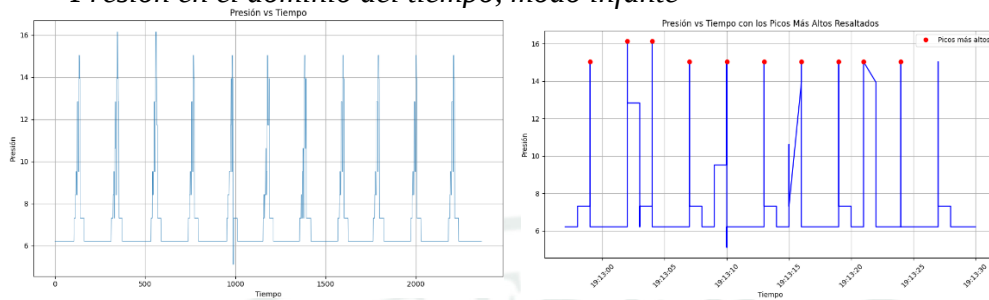


Nota. Elaboración propia.

En Google Colab, se ingresan los datos capturados del modo infantil. Para ello, primeramente, se nombra la cabecera de la tabla como (hora, presión, volumen). Luego, se realiza un filtro para seleccionar los picos más altos de la presión, ya que de esto dependerá la obtención de las "rpm". En la Figura 105 se muestran los picos más altos de la presión con su respectivo tiempo.

Figura 105

Presión en el dominio del tiempo, modo infante



Nota. Elaboración propia.

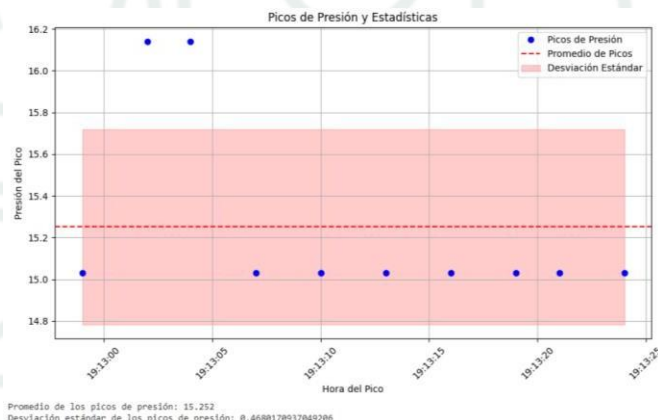
A continuación, se muestra en una tabla los picos más altos, junto con una desviación estándar de 0.46 cmH₂O y un promedio de 15.25 cmH₂O para los picos. Esto es importante para calcular la precisión de la presión y las rpm del ventilador mecánico, como se indica en la Figura 106.

Figura 106

Picos de presión y desviación estándar, modo infante

Tabla de Picos más Altos

Hora del Pico	Presión del Pico
19:12:59	15.03
19:13:02	16.14
19:13:04	16.14
19:13:07	15.03
19:13:10	15.03
19:13:13	15.03
19:13:16	15.03
19:13:19	15.03
19:13:21	15.03
19:13:24	15.03



Nota. Elaboración propia.

La diferencia promedio entre pico a pico resulta ser de 2.7777 segundos. Al dividir 60 segundos entre este valor, obtenemos aproximadamente 21.66 RPM. Según la Sociedad Chilena de Medicina Intensiva, un ventilador debe tener una precisión de 1 cm H₂O, y podemos decir que está dentro de este rango, ya que se estimó un promedio de 22.5 RPM para infantes. Además, el PEEP debe oscilar entre 5 y 25 cmH₂O, lo cual

también está dentro del rango. Podemos interpretar que el prototipo cumple con los requisitos.

5.3.2 *Modo Adulto*

Para el modo adulto se configuró el driver TB6600 en (ON ON OFF ON OFF ON), lo que se traduce en un rango de 3800 y 200 pasos en el código de MicroPython, detallado en la tabla 21. Los resultados de las señales de variación de la presión en el tiempo se obtienen al pulsar el botón "Registro .csv", como se muestra en la figura 107. Este genera archivos con extensión .csv y los almacena en una carpeta del computador con el nombre de "datos", tal como se muestra en la Figura 107. Los datos capturados previamente se pueden visualizar en Excel, con picos que se aproximan a los 20 cmH2O.

Figura 107
Previsualización de datos de rpm con Excel, modo adulto

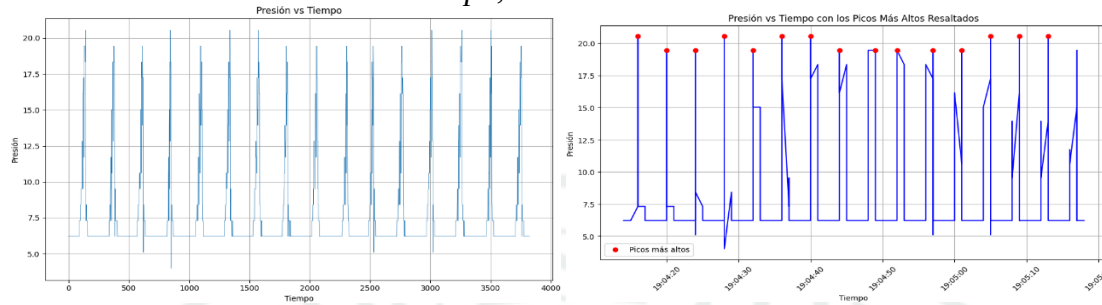


Nota. Elaboración propia.

En Google Colab, se ingresan los datos capturados del modo adulto. Para ello, primeramente, se nombra la cabecera de la tabla como (hora, presión, volumen). Luego, se realiza un filtro para seleccionar los picos más altos de la presión, ya que de esto dependerá la obtención de las "rpm". En la Figura 108 se muestran los picos más altos de la presión con su respectivo tiempo.

Figura 108

Presión en el dominio del tiempo, modo adulto

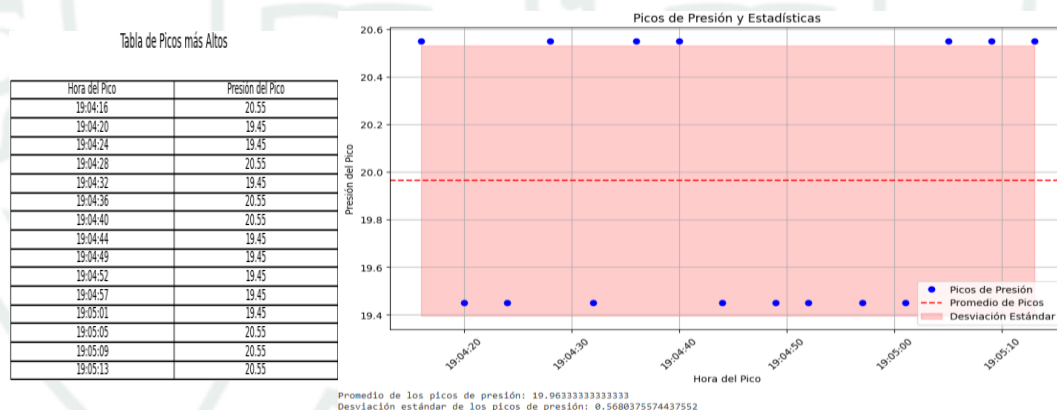


Nota. Elaboración propia.

A continuación, se muestra en una tabla los picos más altos, junto con una desviación estándar de 0.56 cmH₂O y un promedio de 19.96 cmH₂O para los picos. Esto es importante para calcular la precisión de la presión y las rpm del ventilador mecánico, como se indica en la Figura 109.

Figura 109

Picos de presión y desviación estándar, modo adulto



Nota. Elaboración propia.

La diferencia promedio entre pico a pico resulta ser de 4.0714 segundos. Al dividir 60 segundos entre este valor, obtenemos aproximadamente 14.71 RPM. Según la Sociedad Chilena de Medicina Intensiva, un ventilador debe tener una precisión de 1 cm H₂O, y podemos decir que está dentro de este rango, ya que se estimó un promedio de 14 RPM para adultos según el estándar. Además, el PEEP debe oscilar entre 5 y 25 cmH₂O,

lo cual también está dentro del rango. Podemos interpretar que el prototipo cumple con los requisitos.

5.4 Fiabilidad del prototipo

Para evaluar la fiabilidad del prototipo, se realizó una comparación detallada entre sus resultados y características, y los requisitos recomendados por expertos en medicina y algunas instituciones internacionales para los ventiladores mecánicos. Esta evaluación reveló que el prototipo cumple con un 95% de los estándares propuestos y requeridos para la fabricación de ventiladores mecánicos. Esta alta tasa de cumplimiento demuestra la confiabilidad del prototipo y su capacidad para satisfacer las necesidades y expectativas establecidas. En el Anexo L se evidencia la constancia de evaluación de fiabilidad.

Tabla 25
Fiabilidad del prototipo ventilador mecánico

	Ventilador Mecánico (requerimiento)	Prototipo (características)	Calificación
Soporte de estructura (Robustez)	El factor de seguridad estructural debe de ser superior a 1, recomendable 3 veces mayor	El prototipo está construido en material ABS y este cumple con los requisitos.	100%
Intervención manual	Acceso a intervención manual rápida en caso de falla del dispositivo (Sociedad Chilena de Medicina Intensiva)	El prototipo cuenta con control de operación local y a su vez tiene un switch de paro de emergencia.	98%
Interfaz de usuario	Desplegar en interfaz de usuario ajustes valor efectivo: frecuencia respiratoria (Sociedad Chilena de Medicina Intensiva)	El prototipo cuenta con un HMI, donde se puede visualizar la variación de la presión en el tiempo, controlar y capturar datos.	90%

Válvula pasiva	Válvula pasiva de escape para umbral de seguridad. Recomendable: 40 cm H ₂ O defecto o ajustable (Sociedad Chilena de Medicina Intensiva)	El prototipo cuenta con una válvula que no deja pasar el umbral de 40 cm H ₂ O	100%
Lectura en tiempo real	Desplegar en interfaz de usuario Presión de vía aérea (cm H ₂ O). Por lo menos lectura en tiempo real. Precisión deseable 1 cm H ₂ O o mejor. Toma de presión lo más cerca del paciente posible (Sociedad Chilena de Medicina Intensiva)	El prototipo presenta una precisión de lectura de 0.56 cm H ₂ O	100%
Volumen corriente	Volumen corriente entre 200 y 700 ml. Aceptable: 300-600 ml. Continuo o intervalos de 50 ml (Seteable) (Sociedad Chilena de Medicina Intensiva)	El prototipo suministra de 0.42 litros y 0.31 litros para el modo adulto y modo infante	90%
Frecuencia respiratoria	Frecuencia respiratoria (respiración por minuto rpm) 10-40 rpm (Seteable) (Sociedad Chilena de Medicina Intensiva)	El prototipo opera en 14 rpm y 22 rpm	95%
PEEP	PEEP: entre 5 y hasta 20 cm H ₂ O (ideal 25 cm). Precisión: 1 cm H ₂ O (aceptable 2 cm) (Seteable) (Sociedad Chilena de Medicina Intensiva)	El prototipo desde 6 a 24 cmH ₂ O. La precisión es menor a 1 cm H ₂ O	100%

Filtro	Filtros en circuito de inspiración y espiración (Sociedad Chilena de Medicina Intensiva)	El prototipo cuenta con filtro y una válvula para conectar a un tanque de oxígeno	90%
Autonomía de poder	Autonomía de poder mínimo 1 hora: UPS o batería (Sociedad Chilena de Medicina Intensiva)	El prototipo cuenta con una batería de 24V en DC de 2 horas de autonomía	100%
Transportabilidad	Material ligero, acrílicos, PVC, cantidad de piezas reducidas, tamaño no mayor a 70 x 50 x 30 cm, construcción simple, ensamble modular. Una persona no puede transportar o mover objetos mayores a 20 kg (requerimientos de usuario)	El 80% de los materiales empleados en su estructura es de plástico ABS. El peso del prototipo es de 8 kg.	100%
Facilidad de Uso	Interfaz con el usuario sencilla, mecanismo simple, ingreso de pocos Parámetros (Dos pulsadores de control). (requerimientos de usuario)	El prototipo cuenta con 2 pulsadores para seleccionar el modo de operación.	100%
Medición de Presión y Volumen	Sensores electrónicos, Arduino (requerimientos de usuario)	El prototipo cuenta con un sensor analógico MPX5010DP de presión que es capturado por el microcontrolador	90%
Bajos Niveles de Ruido	Motor de tamaño reducido, mecanismo simple, buena lubricación. (requerimientos de usuario)	El prototipo genera ruido de 57dB y 67 dB en los modos de operación	80%

Control de operación	Los ventiladores mecánicos por lo general no cuentan con varios modos de operación.	El prototipo cuenta con 3 modos de operación, alámbrico, inalámbrico y local.	90%
Controlador	Los ventiladores tipo prototipo por lo general son realizados con Arduino.	El prototipo cuenta con el microcontrolador Raspberry Pi Pico y está programado en lenguaje de Python.	90%
Porcentaje de Fiabilidad			94.6%

Nota. Elaboración propia.

5.5 Análisis Económico y Presupuesto del Prototipo

5.5.1 Análisis Económico

El análisis económico para la construcción de este prototipo se fundamentó en los resultados de las encuestas realizadas a expertos en salud e ingeniería, aplicando la técnica conocida como Despliegue de la Función de Calidad, reflejada en la tabla 2 del capítulo 3. La prioridad establecida fue el costo del prototipo, motivo por el cual se optó por emplear materiales de coste reducido para su desarrollo.

5.5.2 Presupuesto del Prototipo

El presupuesto abarca tres aspectos principales. En primer lugar, la adquisición de materiales electrónicos, la mayoría de los cuales se importaron del extranjero, como el sensor de presión, el microcontrolador, el display, entre otros. El segundo aspecto relevante es la construcción de la estructura, utilizando materiales impresos y reciclados. Finalmente, se considera la mano de obra invertida en el diseño, la programación y el ensamblaje del prototipo del ventilador mecánico.

La Tabla 26 detalla los componentes necesarios para construir el ventilador mecánico, así como algunos insumos, especificando sus características y cantidades junto con sus precios respectivos. En cuanto al software utilizado, no se adquirieron licencias, ya que se emplearon versiones para estudiantes como SolidWorks 2020, Visual Basic 2019 y el entorno de desarrollo integrado (IDE) de Micropython, Thonny, de código abierto. En relación con la mano de obra (Tabla 27), se consideró el tiempo invertido por persona en el diseño, la estructura, el circuito, la programación y el ensamblaje del ventilador mecánico.

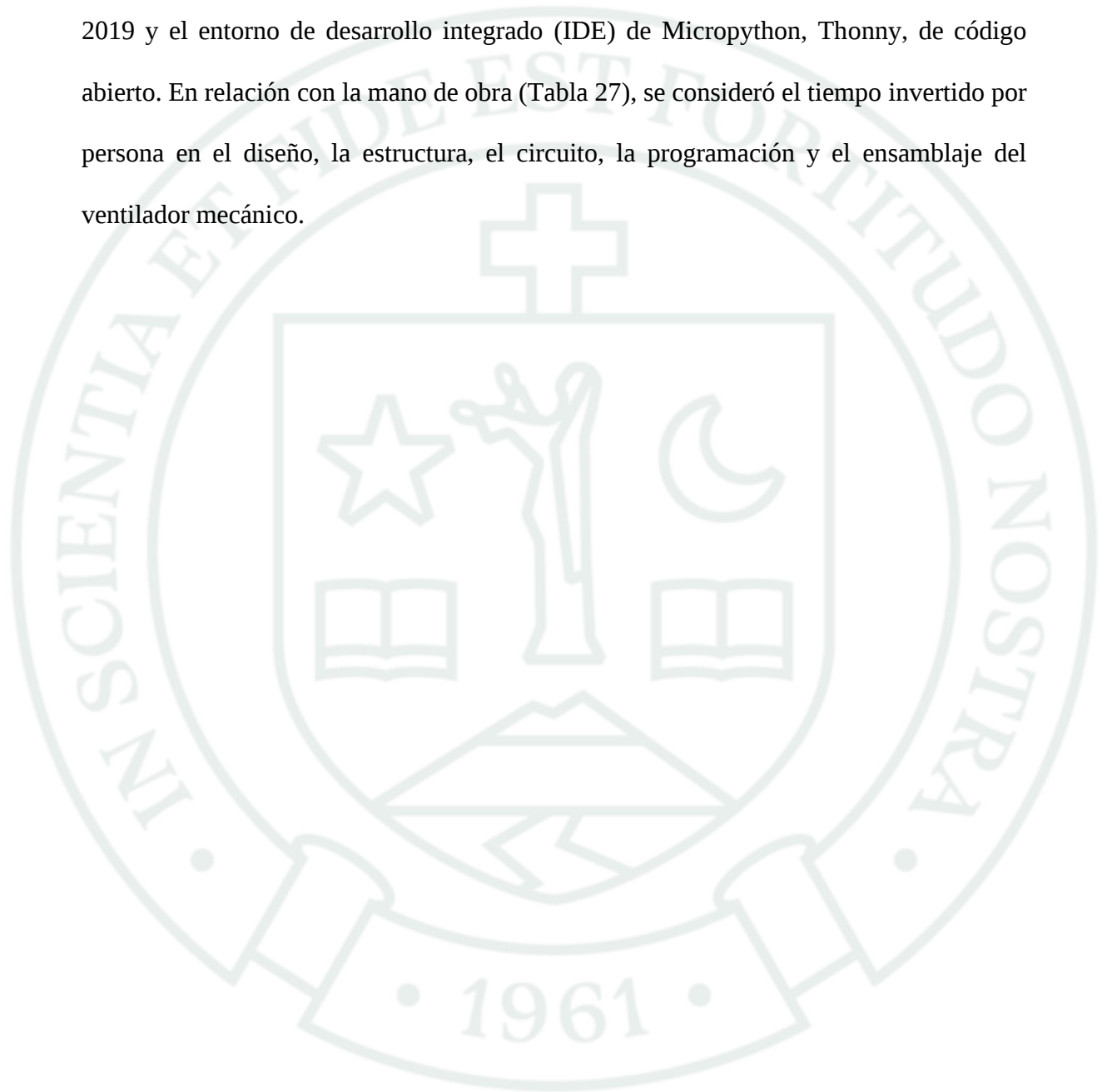


Tabla 26*Presupuesto de materiales del prototipo ventilador mecánico*

NOMBRE	CARACTERÍSTICAS	CANTIDAD	PRECIO
	Acero Inoxidable 304, Paso:		
Tornillo de sin fin	2mm, 4 hilos, avance:8 mm, Longitud:140 mm	1	10.00
Motor Nema 17	Angulo de paso de 1.8 o, 2 fases, corriente de fase de 1,68 A	1	24.00
Rodamientos K08 con soporte a la pared	Diámetro interno de 8 mm, fabricado en Acero	1	3.00
AMBU Resucitador con válvula PEEP	De acción manual, adulto, 2500 ml la bolsa de reservorio	1	30.00
Raspberry Pi Pico	26 pines GPIO multifunción, 2 SPI, 2 I2C, 2 UART, 3 ADC de 12-bit, 16 canales PWM	2	70.00
Driver TB6600 Motor Paso a Paso	Voltaje de operación 9-42 V, Resolución full, 1/2, 1/4, 1/8 and 1/162	1	60.00
Fuente de Alimentación	Conmutada AC 100-240V a DC 12V 5A 60W	1	70.00
Fuente de Alimentación	Para Protoboard de 5V	1	13.00
Fuente de Alimentación	Conmutada AC/DC 100W 24V 4.2A WODE	1	50.00

Sensor de Flujo,	MPX5010DP eléctrica	1	120.00
presión y volumen	transductor de presión		
Display OLED	0.96" SPI 128*64 SSD1306	1	50.00
Piezas 3D	Impresión de la estructura del ventilador con filamento ABS	17	500.00
Guantes Látex	Simulador de pulmón	1	2.00
Regulador de oxígeno	Regulador de flujo Q de 0 a10 L/min para presiones de 15MPa de entrada y 0.2 MPa	1	300.00
Sensor de final de carrera	Voltaje de alimentación: 5V DC Conector 4 cables: 5V, 0V, 0V, SALIDA (ON: 0V, OFF:5V)	1	25.00
Modulo CH-06	UART inalámbrico transceptor	1	25.00
Modulo USB	Bluetooth módulo Bluetooth		
Tubo	USB 3.3V 5V a TTL	1	20.00
Tubo	Transparente, flexible de 1/8 y 1 1/2	2	8.00
Tubo	T	1	5.00
Abrazadera	Metal	2	5.00
ADS1115	Conversor ADC de 16 bits	1	30.00
Botones, resistencias, led,	Insumos del ventilador		10.00
Tuercas, pernos, tornillo y otros	Protección del ventilador mecánico		12.00

Software Licencia	SolidWorks 2020, MIT App Inventor, Visual Basic 2019	3	0
Total			1442.0

Nota. Elaboración propia.

Tabla 27

Presupuesto de mano de obra del prototipo ventilador mecánico

NOMBRE	DESCRIPCION	TIEMPO (h)	Hora Hombre	Total (Soles)
Diseño 3D	Diseño de la estructura del prototipo en SolidWorks	50	15	750
Programación Microcontrolador	Programación de los microcontroladores en Micropython	30	20	600
Programación Interfaz	Programación del interfaz en Visual Basic	20	25	500
Programación App	Programación de una aplicación móvil en MIT App Inventor	15	20	300
Diseño y ensamblado del circuito	Diseño en Easy EDA y ensamblado de forma rustica	20	20	400
Ensamblado del Prototipo	Ensamblado del prototipo en hardware y software	20	10	200
Total			S/. 2750.00	

Nota. Elaboración propia.

CONCLUSIONES

- Se diseñó e implementó un prototipo funcional de ventilador mecánico con capacidad de operación tanto manual como remota, orientado a brindar un soporte en la atención de pacientes con dificultades respiratorias. La implementación fue realizada utilizando software de código abierto y programación en Micro Python, además se desarrolló una interfaz de control y almacenamiento de datos en Visual Basic para los modos de operación de infante y adulto. El prototipo cuenta con tres formas de control, dos de forma remota y una de forma local. Este prototipo está orientado a ser una alternativa viable de asistencia respiratoria, con potencial de optimización futura para mejorar su precisión, autonomía y ser una solución innovadora en el campo de la atención biomédica.
- Se logró desarrollar una comprensión integral del funcionamiento de los ventiladores mecánicos, abarcando el análisis de sus sistemas fundamentales, modos de ventilación, control de parámetros y su interacción con la fisiología respiratoria del paciente. Los ventiladores mecánicos son equipos esenciales en la atención de pacientes con cuidados intensivos ya que les proporcionan un soporte ventilatorio generando una presión positiva que permite el intercambio de gases, además permite ajustar y monitorear parámetros como la presión, el flujo, la humedad, concentración de oxígeno adaptándose a las necesidades del paciente con la finalidad de evitar posibles complicaciones en la recuperación del paciente. Su uso adecuado puede mejorar significativamente la supervivencia y calidad de vida de los pacientes en estado crítico. Este conocimiento que se encuentra en las bases teóricas permitió identificar cómo el equipo asiste en el proceso respiratorio del paciente.

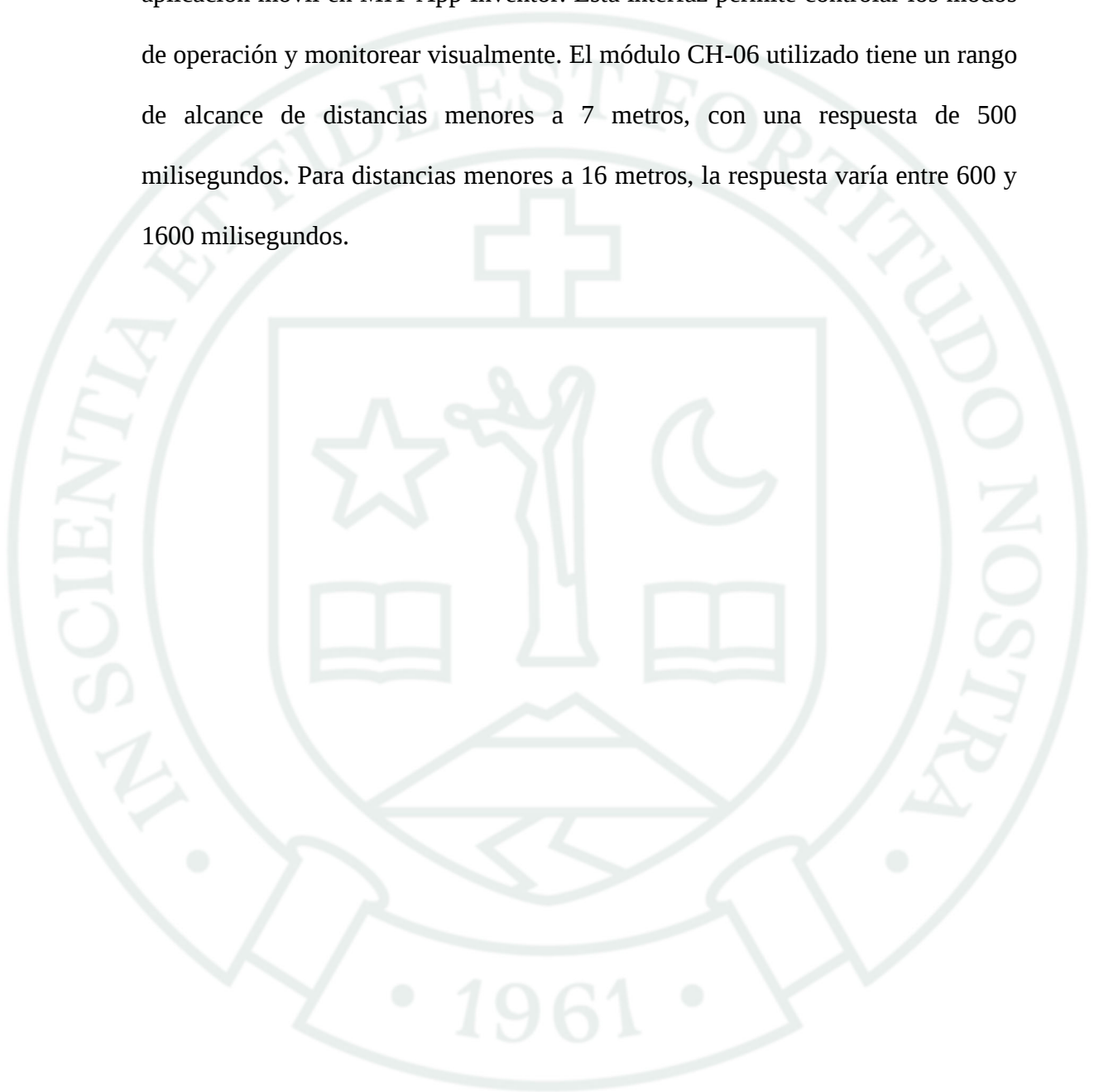
- Se consiguió determinar los requisitos fundamentales del ventilador mecánico mediante el análisis de las necesidades clínicas, parámetros técnicos esenciales y las consultas realizadas a especialistas en temas relacionados con ventiladores mecánicos, quienes nos dieron ideas de las funciones que debe cumplir nuestro prototipo, por ejemplo, respecto a la parte médica nos indicaron que se deben obtener parámetros como la presión y el volumen, los cuales deben ser monitoreados constantemente. En la parte técnica, las dimensiones que debería tener el equipo para ser lo más portátil posible. Por nuestra parte, aportamos nuestros conocimientos en hardware y software tomando en cuenta que el prototipo trata de ser un equipo accesible para todos los que lo necesiten, cabe resaltar que también se pudo apreciar algunas carencias en el manejo de este tipo de equipos y en el mantenimiento por lo cual tratamos de que el equipo sea de fácil uso y mantenimiento.
- Se seleccionaron los componentes idóneos para el diseño, mediante un proceso de evaluación técnica basado en la compatibilidad, disponibilidad, durabilidad y cumplimiento de los requerimientos funcionales previamente establecidos también se priorizó el uso de materiales ligeros, compactos y la integración de tecnología tanto en software y hardware reduciendo significativamente el peso total. Esta selección consideró aspectos esenciales como la durabilidad, eficiencia energética e integración entre los subsistemas electrónicos, mecánicos y de control, garantizando un funcionamiento coherente y articulado dentro del diseño propuesto. Los resultados obtenidos evidencian que la elección de componentes constituye un factor determinante para la portabilidad y durabilidad del prototipo, consolidándose como una base sólida para las etapas posteriores de implementación. De este modo, el proceso efectuado contribuye al desarrollo de

una solución técnica consistente, alineada con los objetivos de la tesis y con potencial de aplicación en el campo de los sistemas de asistencia respiratoria.

- La ISO 10651-4 es parte de la serie de normas ISO que trata sobre ventiladores pulmonares, y específicamente, la ISO 10651-4 aborda los requisitos técnicos y de funcionamiento para los ventiladores pulmonares que se utilizan fuera de los entornos hospitalarios convencionales, como los ventiladores de emergencia y de transporte. En cuanto al cumplimiento de la norma ISO 10651-4 se cumplieron los requisitos generales de diseño y fabricación que se refieren a la portabilidad y durabilidad del prototipo y requisitos de funcionalidad tales como los modos de control y funciones básicas para el monitoreo del volumen y presión del prototipo
- Se evaluó la confiabilidad y el desempeño del prototipo a lo largo del tiempo mediante pruebas y análisis orientados a verificar la estabilidad de funcionamiento, resistencia operativa y consistencia en su rendimiento bajo condiciones prolongadas de uso. Los resultados obtenidos permitieron determinar que el dispositivo presenta un comportamiento estable, con una adecuada capacidad de respuesta y un nivel de desempeño que cumple con los requerimientos establecidos, lo cual garantiza su durabilidad y fiabilidad operativa. Este proceso de evaluación es fundamental, ya que, valida la capacidad del sistema para mantener un funcionamiento continuo, minimizando riesgos de falla y asegurando su uso en escenarios reales, lo que contribuye a la consolidación del prototipo como una solución técnica confiable.
- Se ha desarrollado con éxito la conexión alámbrica mediante USB serial entre el prototipo y una interfaz de operación remota diseñada en Visual Basic. Desde esta interfaz, se pueden controlar los dos modos de operación del ventilador mecánico

y se visualiza la presión mantenida en cada modo.

- Se ha desarrollado con éxito la conexión inalámbrica mediante la comunicación Bluetooth entre el prototipo y una interfaz de operación, desarrollando una aplicación móvil en MIT App Inventor. Esta interfaz permite controlar los modos de operación y monitorear visualmente. El módulo CH-06 utilizado tiene un rango de alcance de distancias menores a 7 metros, con una respuesta de 500 milisegundos. Para distancias menores a 16 metros, la respuesta varía entre 600 y 1600 milisegundos.



RECOMENDACIONES

- Se recomienda utilizar las nuevas versiones de microcontroladores, como las versiones 2022 W, ya que incluyen conectividad WIFI y BLUETOOTH. Además, es probable que estos microcontroladores trabajen con dos núcleos, ya que en el presente trabajo los microcontroladores actuales presentan fallas y por eso se utilizaron dos microcontroladores.
- Se recomienda utilizar conversores analógico-digitales (ADC) como el ADS1115, ya que permiten aumentar la resolución de la señal analógica. En el presente trabajo, fue útil utilizar este tipo de ADC para aumentar la resolución en bits y obtener gráficos más precisos de la presión y las señales de flujo.
- Es posible mejorar la automatización de los ventiladores mecánicos en los hospitales, también es recomendable incorporar la funcionalidad de almacenamiento de datos o registro de los parámetros. Esto permitirá recopilar información sobre el funcionamiento del ventilador, como la presión, la frecuencia respiratoria y otros datos relevantes. El registro de estos parámetros puede ser útil para el seguimiento de los pacientes, el análisis de tendencias y la toma de decisiones médicas.
- En relación a la formación académica en electrónica, es altamente recomendable ampliar los conocimientos en programación utilizando diferentes lenguajes y explorar el uso de diferentes microcontroladores. Esto permitirá aumentar la capacidad de solución en ingeniería, incluyendo el desarrollo y mejora de ventiladores mecánicos. La programación es una habilidad fundamental en el campo de la electrónica y conocer distintos lenguajes y microcontroladores brinda flexibilidad y versatilidad para abordar diferentes desafíos y optimizar el

funcionamiento de los dispositivos.

- Los ventiladores mecánicos requieren sistemas de control sofisticados para regular parámetros como la presión y el volumen. Es crucial que los estudiantes reciban formación en control automático, retroalimentación y sistemas de regulación basados en microcontroladores y controladores lógicos programables, también amplificación de potencia y fuentes de alimentación estables que se utilizan para controlar los motores y actuadores del ventilador mecánico y los sensores que monitorean la presión y el flujo ya que estos datos se utilizan para ajustar automáticamente los parámetros del ventilador.
- La formación en Ingeniería Electrónica debe adaptarse para abarcar tanto los aspectos técnicos como normativos que exigen los equipos médicos. Además, debe promover el desarrollo de habilidades multidisciplinarias que permitan a los ingenieros colaborar de manera efectiva con áreas de trabajo, por lo cual sería muy bueno incluir en el programa académico formación sobre las normativas específicas para dispositivos, como la ISO 10651-4 para ventiladores, y otras regulaciones locales e internacionales. Los estudiantes deben comprender los procesos de certificación, pruebas de seguridad y protocolos de validación que garantizan la seguridad del usuario.

BIBLIOGRAFÍA

Abad, B. (2015). Desarrollo e implantación de una placa de test gestionada por Arduino.

Al Hussein, A. M., Ju Lee, H., Negrete, J., & Powelson, S. (2010). Design and Prototyping of a Low-cost Portable Mechanical Ventilator. USA: Proceedings of the 2010 Design of Medical Devices Conference .

alldatasheet.com. (2012). *MPX5010DP*. Obtenido de <https://www.alldatasheet.com/view.jsp?Searchword=MPX5010DP&sField=4>

Carpio, S., & Sughey, E. (2022). Diseño e Implementación de Prototipo de ventilador mecánico de bajo costo utilizando una placa programable y software libre para monitoreo de parámetros mínimos necesarios para dar soporte vital a pacientes con deficiencia respiratoria aguda. *Ug.edu.ec*. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/59828>

César, N., Flor , O., Tapia , J., Flores , E., Coba , A., & Chango , E. (Mayo de 2020). Diseño de ventilador mecánico emergente en modo asistido/controlado y espontáneo por presión. Universidad de las Américas, Ingeniería en Redes y Telecomunicaciones.

Ciberes.org. (2020). *Crean un respirador de bajo coste para zonas con pocos recursos* | CIBERES. Obtenido de <https://www.ciberes.org/noticias/crean-un-respirador-de-bajo-coste-para-zonas-con-pocos-recursos>

CNA. (2001). *Serie Autodidáctica De Medición Tubos Venturi, Dall Y Tobera Coordinación De Tecnología Hidráulica (Imta) Subdirección General De Administración Del Agua (Cna) C É IMTA*. Subdirección General De Administración Del Agua, Mexico. Obtenido de

http://repositorio.imta.mx/bitstream/handle/20.500.12013/1174/IMTA_083.pdf?sequence=1&isAllowed=y

Díaz Rivadeneira, S. P. (Agosto de 2015). Ensamble y programación de un prototipo de respirador artificial de bajo costo con tres modos de operación. Ecuador: UNIVERSIDAD SAN FRANCISCO DE QUITO.

Equipo de Proyecto UFG y Asociados. (24 de Abril de 2020). Ventilador mecánico pulmonar EJÉKAT-V1.0. Universidad Francisco Gavidia.

Firma Ambu S.L. (2015). *Catálogo Emergencias*. Obtenido de https://www.ambu.es/Admin/Public/Download.aspx?file=Files%2FFiles%2FDownloads%2FAmbu+ES%2FES_Cat%C3%A1logo_Emergencias.pdf

Flórez Bautista, E. (Junio de 2020). Diseño para la elaboración de un prototipo de ventilador mecánico de emergencia por Covid 19 en la Clínica Misericordia Internacional de la ciudad de Barranquilla. Universidad Militar Nueva Granada.

Heredia, O., Chunga, X., De La Cruz, L., & Zimic, M. (2021). Diseño y evaluación de un ventilador mecánico. SciELO Preprints.

Khan Academy. (2022). *Khanacademy.org*. Obtenido de <https://es.khanacademy.org/science/physics/fluids/fluid-dynamics/a/what-is-bernoullis-equation>

Kshetry, R., Gupta, A., Chattopadhyaya, S., & Srivastava, M. (2021). Design and Analysis of a Low-Cost Electronically Controlled Mobile Ventilator, Incorporating Mechanized AMBU Bag, for Patients during COVID-19 Pandemic. *Journal of Healthcare Engineering*, 15. Obtenido de <https://doi.org/10.1155/2022/6436818>

MD Dispositivos Medicos del Peru. (s.f). Regulador para balon de oxigeno. Obtenido de https://www.mdperu.com/wp-content/uploads/2022/07/REGULADOR-DE-OXIGENO_V2.pdf

Perez Correa, P. (2020). Sistema de medida del angulo de ataque mediante sensores de presion diferencial. pág. 20. Obtenido de [https://riunet.upv.es/bitstream/handle/10251/148504/P%
c3%a9rez%20%20Sistema%20de%20medida%20del%20%20c3%a1ngulo%20de%20ataque%20mediante%20sensores%20de%20presi%3%b3n%20diferencial.pdf?sequence=1&isAllowed=y](https://riunet.upv.es/bitstream/handle/10251/148504/P%c3%a9rez%20%20Sistema%20de%20medida%20del%20%20c3%a1ngulo%20de%20ataque%20mediante%20sensores%20de%20presi%3%b3n%20diferencial.pdf?sequence=1&isAllowed=y)

Romero , P., Lázaro, C., & González, J. (s.f.). Estadística descriptiva e inferencial. Obtenido de https://www.researchgate.net/profile/Pedro-Romero-Aroca/publication/275021043_Estadistica_Descriptiva_e_Inferencial/links/55bfd42b08aec0e5f4476a2a/Estadistica-Descriptiva-e-Inferencial.pdf

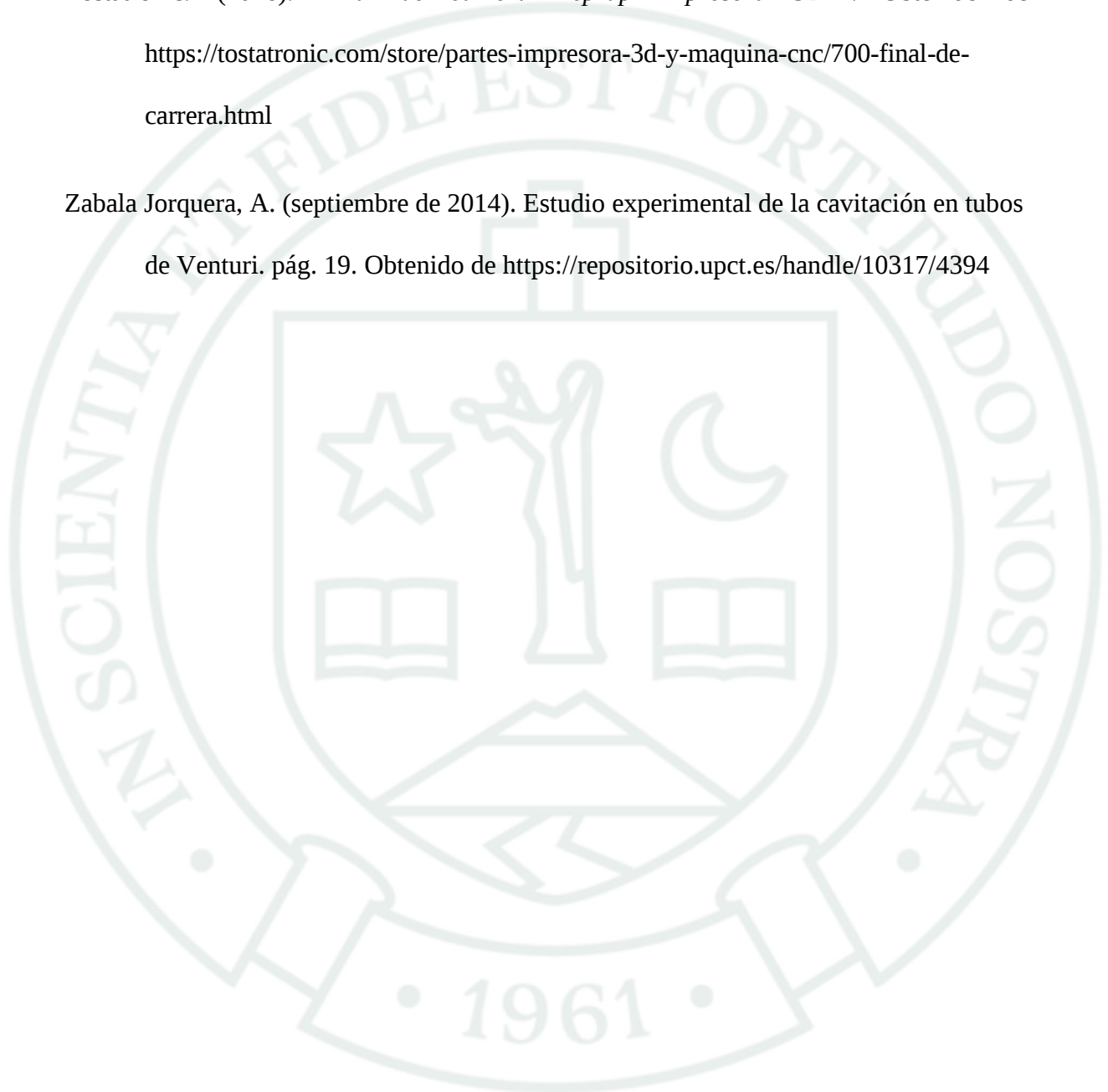
Ruiz, A., Mejía, A., & Gaviria, N. (10 de 12 de 2018). Aplicación de metodología design thinking en el desarrollo de cortadora automática CNC para MiPyME de confección. págs. 157-168. Obtenido de <https://www.redalyc.org/journal/5537/553762534017/553762534017.pdf>

Sanchez, V. (2021). *Tras el Covid-19, científicos del mundo se preparan para una futura pandemia.* Obtenido de <https://www.france24.com/es/programas/ciencia-y-tecnolog%C3%ADa/20210326-pandemias-futuras-coronavirus-sars-mers-cientificos>

Spectra Premiun. (s.f). *Sensor de flujo de aire* Obtenido de
<https://www.spectrapremium.com/es/aftermarket/north-america/mass-air-flow-sensor>

Tostatronic. (2016). *Final de carrera Reprap Impresora 3D* . Obtenido de
<https://tostatronic.com/store/partes-impresora-3d-y-maquina-cnc/700-final-de-carrera.html>

Zabala Jorquera, A. (septiembre de 2014). Estudio experimental de la cavitación en tubos de Venturi. pág. 19. Obtenido de <https://repositorio.upct.es/handle/10317/4394>



ANEXOS

Anexo A: Ventiladores Respiradores, Perú

Respirador Newport e360: es uno de los ventiladores más usados en el Perú.

Respirador Newport™ e360 – una mirada más profunda

1. Facilidad en la identificación de violaciones de alarma gracias a su lámpara de alarma de 360° de visualización y a su pantalla de mensajes de alarma inteligente.
2. Los datos monitoreados en detalle y las funciones avanzadas se fusionan en las pantallas de fácil navegación de la interfaz gráfica del usuario.
3. El modelo Newport e360T* permite la conexión de la GUI integrada a una monitor táctil externo más grande.
4. Configuración fácil y rápida por el acceso directo a los controles de ventilación estándar.
5. El método de aceptación de un simple toque para todos los ajustes ayuda a prevenir los cambios accidentales.
6. La función no invasiva integrada amplía la versatilidad de la aplicación con una avanzada compensación de fugas.

*Otros modelos pueden utilizar un monitor externo de pantalla plana opcional para visualización remota



ESPECIFICACIONES DEL RESPIRADOR NEWPORT™ e360

NOTA: El respirador Newport™ e360 está disponible en diversas configuraciones para ajustarse a sus necesidades clínicas. Contacte su representante local para más información.

RANGO DE PACIENTE

Usando pescozo* - adulto

TIPOS/MODOS DE RESPIRACIÓN

Control de volumen		
Control de presión	Ventilación de liberación de presión "bifásica" (BPRV)	Opciones de respiración espontánea: Presión de soporte, Presión de soporte con volumen objetivo*
Control de presión con volumen objetivo*	A/C/MV SIMV ESPONT	NIV (ventilación no invasiva) Encendido/Apagado en todos los tipos/modos de respiración

CONTROLES/AJUSTES

Automático/Control	Tasa de resp. (frecuencia)	Relación I:E	Insp. mantenida
Pendiente/Crecimiento ajustable	Patrón de onda de flujo	Trigger (sensibilidad) por presión o flujo	Exp. Mantenido NIFP D1
Umbral espiratorio automático o ajustable	Pausa Suspiro	PEEP/CPAP	Manobra PV +
(FlexCycle)	Límite de presión	Compensación de flujo base	O2 (3 min)
Volumen corriente	Tiempo inspiratorio		Insuflación manual
Flujo			

PARAMETROS MONITOREADOS

PiCO	C-Bias	Con. espira	Tiempo inspiratorio
% de varianza VTE	WOBs/MVE	Flujo exp.	PEEP
RR/respont.	Constante de tiempo	PiPai	MVE/respont.
MVI	PEEP total	VTI	PIO
Relación I:E	Flujo insp.	RSB	VTE
Pmedia		IRI	RRast

Cada respirador e360 incluye:

ALARMAS AJUSTABLES

Volumen/minutador de espira/baja y alta	Presión en la vía aérea baja y alta
Tasa de resp. alta	Desconexión (% umbral)
Apnea	

ALARMAS AUTOMÁTICAS

[*]Muestro liberación de rango PtiO ₂ bajo y alto	Presión de línea base alta sostenida	Temporizador de inspiración muy largo
Límite de la presión por debajo de la PEEP	Tiempo insp. muy corto	Baja y alta presión de línea de base (PEEP)
Paw baja por debajo de la PEEP	Violación a la relación inversa I:E	Volumen objetivo no alcanzado

DIMENSIONES (SOLO EL RESPIRADOR)

Ancho: 13,9 pulg. (35,3 cm)	Altura: 12,1 pulg. (30,7 cm)
Profundidad: 13,9 pulg. (35,3 cm)	Peso: 38 lb (17,3 kg)

Modelos EE.UU. – Newport e360S-EE.UU., e360T-EE.UU.

Modelos OUS (fuera de los EE.UU.) – Newport e360S, e360E, e360T

Para obtener un listado completo de las especificaciones remítase a la ficha de especificaciones de cada respirador Newport e360 disponible en Covidien.com/RM6.

*No disponible para el modelo OUS Newport e360S

†Únicamente disponible para el modelo OUS Newport e360E, T

+Únicamente disponible para el modelo OUS Newport e360T

Para obtener información detallada de piezas y pedidos, contacte a su representante de Covidien

INFORMACIÓN DE PEDIDO Únicamente fuera de los EE.UU.

E360TH-XX-YY	Respirador Newport e360TH (únicamente la unidad principal), monitor táctil externo compatible y características premium adicionales
E360E-XX-YY	Respirador Newport e360E con especificaciones expandidas y características premium adicionales
E360S-XX-YY	Respirador Newport e360S con especificaciones y características estándar
E360T-XX-YY	Respirador Newport e360T con monitor de pantalla táctil externo, brazo de montaje y características premium adicionales

Cada respirador Newport e360 incluye:

Válvula de exhalación caliente reutilizable integrada con dos sensores de flujo y tubos de oxígeno y aire.

Kit de accesorios estándar

Soporte para el brazo de extensión: BKT2199P

Gancho para el circuito de respiración del paciente: HQR210P

Montaje para el brazo de extensión: EXT450P

Filtro antibacteriano desechable (dos) FLT1801Q

Escoja entre los siguientes idiomas de los rótulos del panel POC

XX - Idioma de los rótulos del panel

WW - inglés	CN - chino	JP - japonés	RU - ruso
SY - símbolos	ES - español	PT - portugués	VT - vietnamita

Escoja entre los siguientes estilos de cables de alimentación eléctrica (YY)

YY - Tipo de cable eléctrico

NA - Norteamérica IN - Europa UK - Reino Unido BR - Brasil

INFORMACIÓN DE PEDIDO Únicamente en EE.UU.

E360T-US-NA	Respirador Newport e360T con monitor de pantalla táctil externo, brazo de montaje y características premium adicionales. Versión para EE.UU.
E360TH-US-NA	Respirador Newport e360TH (únicamente la unidad principal), monitor de pantalla táctil externo con características premium adicionales y compatibles versión para EE.UU.

E360S-US-NA Respirador Newport e360S con especificaciones y características estándar versión para EE.UU.

Cada respirador Newport e360 incluye:

Válvula de exhalación caliente reutilizable integrada con dos sensores de flujo

Kit de accesorios estándar

Freno para el brazo de extensión: BKT2199P

Gancho para el circuito de respiración del paciente: HQR210P

Montaje para el brazo de extensión: EXT450P

Filtro antibacteriano desechable (dos)

FLT1801Q

Cable eléctrico para Norteamérica

PWR1500P

Tubo de aire a alta presión transparente

de 10 pies/3 m con conector D68 en

ambas terminaciones: H06801P

Tubo de oxígeno a alta presión verde de 10

pies/3 m con conector D68 en ambas

terminaciones: H06803P

ACCESORIOS

Monitor de pantalla plana táctil de 13" *** grado médico, brazo de soporte articulado y equipo de montaje, cable eléctrico para Norteamérica	FPD2126A
Monitor de pantalla plana táctil de 13" *** grado médico, brazo de soporte articulado y equipo de montaje, cable eléctrico de tipo internacional	FPD2126A-IN
Respirador Newport e360S con especificaciones y características estándar	FPD2120A
Monitor de pantalla plana táctil de 17" *** grado médico, brazo de soporte articulado y equipo de montaje, cable eléctrico de tipo internacional	FPD2120A-IN
Carro compacto para respirador Newport e360 con soporte de montaje empotrado para humidificador	CRT360A
Compresor de aire con carro de soporte para el respirador Newport e360	
Compresor de aire 120V/60Hz C235	C235-120V-60-1
Compresor de aire 220V/50Hz C235	C235-230V-50-1
Compresor de aire 220V/60Hz C235	C235-230V-60-1
Kit calentador del filtro antibacteriano con colector de agua, tubos del circuito y conector de silicona de 22 mm de diámetro externo	
Kit calentador del filtro antibacteriano para 100-120V	KIT2110A
Kit calentador del filtro antibacteriano para 220-240V	KIT2220A

***Únicamente para ser utilizado con los modelos T o TH



Oficinas

Covidien Colombia S.A.
Edificio Prados de la Morena
Carretera Central del Norte (Cra 7 a)
Kilómetro 18 / Chía, Cundinamarca
Colombia
[t] (57-4) 668 3777
[f] (57-4) 668 3777 Ext. 105

Covidien Perú S.A.
Jr. Monterrey 341 Of. 302,
Urb. Chacarilla del Estanque
Santiago de Surco, Lima, Perú
[t] (51-1) 611 8700 Ext. 23
[f] (51-1) 611 8715

Distribuidores Autorizados

Venezuela
Equisa
R.L.F. J-00049120-8
Avda. Francisco Solano López
C.C. Chacaito PB. Local 103 Equisa
Caracas, Venezuela
[t] (58) 212 952 1084 / 952 1659

Ecuador
ALEM CIA LTDA
La isla N31 - 12 y San Gabriel
Quito, Ecuador
[t] (593) 2 255 5839
[f] (593) 2 255 2816

Perú
Premium Medical E.I.R.L.
Calle Ricardo Angulo Ramírez No. 695
San Isidro, Lima 27, Perú
[t] (51-1) 226 8175 / (51-1) 226 4280
[f] (51-1) 226 8175

Respirador Avante Verdena MC:

Avante Verdana MC

ICU Ventilator

The Avante Verdana MC respiratory ventilator is built with patient safety and ease of use in mind. It includes both automatic and manual controls. The Verdana MC can be used for control ventilation, support ventilation, and spontaneous respiration/CPAP. This ventilator is easily transportable, and features an ergonomically designed user interface.

The Avante Verdana MC ventilator is available through Avante Health Solutions. For more information, contact an Avante representative today.

FEATURES

- › Can be used on infants, pediatrics, and adults
- › 12.1-inch color LCD display screen
- › Touch screen interface
- › Backup ventilation to ensure patient safety by preventing suffocation
- › Monitoring functions for respiratory dynamics include pressure-volume loop, volume-flow loop, and pressure-flow loop
- › Physiological and technical alarms



SPECIFICATIONS



Ventilator Weight:
27.6 lbs (12.5 kg)

Trolley Weight:
44.1 lbs (20 kg)



Ventilator Dimensions:
17.9" x 15.7" x 11.8"
(45.5 x 40 x 30 cm)

Trolley Dimensions:
22.2" x 21.1" x 35.8"
(56.5 x 53.5 x 91 cm)



Power:
100-240V,
50/60 Hz



Respiratory Modes:
VCV, PCV, SIMV-VC+ PSV,
SIMV-PC+PSV,
SPONT (CPAP/PSV),
BIPAP, APRV, V

Optional Modes:
PRVC, NIV-PC
NIVCPAP, SIMV-PRVC

2366-2020-07-13

2601 Stanley Gault Pkwy, Suite 101 • Louisville, KY 40223 USA
800-477-2006 • 502-244-4444 • FAX: 502-244-0369

www.avantehs.com

Avante Health Solutions

SPECIFICATIONS (cont.)

Tidal Volume

Range:

Adult: 100 - 2500 mL

Pediatric: 50 - 380 mL

Neonatal: 20 - 100 mL

Increment:

10ml, or 5 ml for values between
20-100ml

Breathing System

VBS Compliance: < 4mL/cmH₂O

Recommended Breathing
System Compliance: < 50mL/cmH₂O

Recommended Breathing
System Resistance:
< 3cmH₂O @ 60L/min

Inspiratory Channel

Flowrate 60L/min:
6cmH₂O maximum

Flowrate 30L/min:
3cmH₂O maximum

Flowrate 5L/min:
1cmH₂O maximum

Inspiratory Resistance:
6cmH₂O at 60L/min

Gas Transportation Flowrate Range:
0-180L/min

Expiratory Channel

Flowrate 60L/min:
< 2cmH₂O maximum

Flowrate 30L/min:
< 1cmH₂O maximum

Flowrate 5L/min:
< 0.5cmH₂O maximum

Expiratory Resistance:
3cmH₂O @ 60L/min

Expiratory Flowrate Measuring Range:
0-180L/min



2801 Stanley Gault Pkwy, Suite 101 • Louisville, KY 40223 USA
800-477-2006 • 502-244-4444 • FAX: 502-244-0369

www.avantehs.com

Avante Health
Solutions

Anexo B: Características y hoja de datos de los Componentes

En este apartado se mencionan las importantes características de los componentes biomédicos y electrónicos.

Anexo B1: Bolsa Ambu

Ambu® SPUR II

Ambu SPUR II representa una familia completa de reanimadores, y comprende tamaños para neonatos, pediátricos y adultos. Ambu SPUR II ofrece a los usuarios una excepcional sensación táctil y visual durante la reanimación. La bolsa es muy sensible y muestra una resistencia mecánica mínima. Además, su diseño característico proporciona un volumen de expulsión óptimo con un retroceso perfecto.

Todos los reanimadores Ambu SPUR II se presentan en bolsas de plástico para transporte individuales y desechables, con una o más mascarillas y los accesorios especiales. Las bolsas tienen un código de colores para identificar rápidamente los tamaños para neonatos, pediátricos y adultos.

Datos Técnicos

SPUR II Adultos - diseñados para peso a partir de 30 kg (10 años)
Máx. Volumen tidal aprox: una mano 800 ml dos manos 1.100 ml
Dimensiones (longitud x diámetro): 295 x 127 mm
Conector del paciente:
Externo 22 mm macho (ANSI/ISO)
Interno 15 mm hembra (ANSI/ISO)
Volumen O₂ Reservorio aprox: 2.600 ml
Conector espiratorio: 30 mm
Válvula limitadora de presión: 40 cm H₂O

SPUR II Pediátrico - diseñado para peso aprox. de 10-30 kg. (1- 10 años)
Máx. Volumen tidal aprox: 450 ml
Dimensiones (longitud x diámetro): 234 x 99 mm
Conector del paciente:
Externo 22 mm macho (ANSI/ISO)
Interno 15 mm hembra (ANSI/ISO)
Volumen O₂ Reservorio aprox.: 2.600 ml
Conector espiratorio: 30 mm
Válvula limitadora de presión: 40 cm H₂O

SPUR II Neonatos - diseñados para pesos por debajo de 10 kg (1 año)
Máx. Volumen tidal aprox: 150 ml
Dimensiones (longitud x diámetro): 168 x 71 mm
Conector del paciente:
Externo 22 mm macho (ANSI/ISO)
Interno 15 mm hembra (ANSI/ISO)
Volumen O₂ Reservorio aprox.: 300 ml
Conector espiratorio: 30 mm
Válvula limitadora de presión: 40 cm H₂O

Características y Beneficios

- **Funcionalidad**
Sistema de válvula con un sólo obturador para un funcionamiento seguro.
- **Reduce la fatiga durante la ventilación manual**
La tira de soporte integrada asegura una fácil sujeción para mayor comodidad y presión uniforme, lo que ayuda a reducir la fatiga durante la ventilación manual; también para los usuarios con manos pequeñas.
- **Buena sensación de compliancia pulmonar**
El diseño único de doble pared permite al usuario una buena sensación de la presión de ventilación y compliancia pulmonar del paciente.
- **Agarre Seguro**
La Superficie SafeGrip™ (Superficie Segura) asegura que los usuarios puedan mantener una firme sujeción en situaciones estresantes.
- **Permite medicación oral**
El puerto de medicación permite la administración rápida de medicamentos sin desconectar el Ambu SPUR II del tubo ET, lo que permite suministrar medicación oral sin tener que retirar las máscara facial ni parar la ventilación.

ESPECIFICACIONES PARA PEDIDO - SPUR II

	REF.
SPUR II Adulto , con bolsa recipiente de O ₂ y mascarilla desechable con válvula (medium n°5) / 12 piezas por caja	325 002 000
SPUR II Adulto , con válvula limitadora de presión, bolsa recipiente de O ₂ y mascarilla desechable con válvula (medium n°5) / 12 piezas por caja	325 012 000
SPUR II Pediátrico , con bolsa recipiente de O ₂ y mascarilla desechable con válvula (pediátrico n°2) / 12 piezas por caja	330 003 000
SPUR II Pediátrico , con bolsa recipiente de O ₂ y mascarilla desechable con válvula (toddler n°3) / 12 piezas por caja	330 004 000
SPUR II Neonato , con tubo recipiente de O ₂ y mascarilla desechable con válvula (neonato n°1) / 12 piezas por caja	335 002 000
SPUR II Neonato , con bolsa recipiente de O ₂ y mascarilla desechable con válvula (neonato n°1) / 12 piezas por caja	335 102 000
SPUR II Neonato , con bolsa recipiente de O ₂ y mascarilla desechable con válvula (pediátrico n°2) / 12 piezas por caja	335 103 000



* 12

DATOS TÉCNICOS

Las versiones adulto y pediátrico generan volumen tidal de 700ml y 450 ml respectivamente.

El Ambu Resucitador Silicona Oval no contiene látex.

La bolsa está hecha de silicona.

Ambu Resucitador Silicona Oval, adulto, peso corporal desde 30kg (10 años)

Volumen 1475 ml. Dimensiones (largo x diámetro) 291 mm x 128 mm (11.45 pulg. x 5 pulg.)

Ambu Resucitador Silicona Oval, pediátrico, peso corporal de 10-30 kg (1 - 10 años).

Volumen 635 ml. Dimensiones (largo x diámetro) 245 mm x 99 mm (9.65 pulg. x 3.9 pulg.)

Ambu Resucitador Silicona Neonato, hasta 10 kg

Volumen corriente max. de aprox. 150 ml. Dimensiones: (longitud x diámetro) 165 x 70 mm

Conector del paciente: 22/15 mm (ISO). Volumen del recipiente de O₂: aprox. 100 ml

Volumen total del resucitador: 220 ml. Conector espiratorio: 18 mm

Conector de manómetro: sí

Ambu Resucitador Silicona Oval se puede esterilizar en autoclave a 134°C

El volumen de la bolsa del depósito de oxígeno es de aproximadamente 2600 ml.

Conector del paciente: Exterior 22 mm (ISO)

Interior 15 mm (ISO)

Conector espiratorio (para válvula PEEP): 30 mm macho (ISO)

Fugas (ambas direcciones) No se puede medir

Temperatura de funcionamiento: -18 °C a 50 °C (-4 °F a 122 °F)

con una humedad entre 15% y 95%

Temperatura de almacenamiento: -40 °C a 60 °C (-40 °F a 140 °F)

con una humedad entre 40% y 95%



La versión adulto está disponible con y sin válvula limitadora de presión a 40 cm² con tapa de anulación.

ESPECIFICACIONES DEL PEDIDO

REF.

Ambu Resucitador Silicona Oval, Adulto, con válvula de paciente, bolsa reservorio de O ₂ y mascarilla de silicona tamaño 5.	470 003 000
Ambu Resucitador Silicona Oval, Adulto, con válvula de paciente, válvula limitadora de presión, bolsa reservorio de O ₂ y mascarilla de silicona tamaño 5.	470 003 100
Ambu Resucitador Silicona Oval, Pediátrico, con válvula de paciente, válvula limitadora de presión, bolsa reservorio de O ₂ y mascarilla de silicona manguito abierto tamaño 2.	370 003 000
Ambu Resucitador Silicona Oval, Neonato con válvula de paciente de cierre individual para neonatos con pivote. Válvula que limita la presión 40 cm H ₂ O y puerto de manómetro.	
Tubo de oxígeno abierto. Mascarilla de silicona manguito abierto. Tamaño mascarilla 0.	288 103 000

Subsidiaria

Firma Ambu S.L

Alcalá, 261-265, Edif. 2-3ª dcha.

28027 Madrid

T: (+34) 91 411 68 30

F: (+34) 91 564 50 82

www.ambu.es

Fabricante

Ambu A/S

DK-2750 Ballerup

www.ambu.com



US: Rx only

Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

The MPX5010/MPXV5010G series piezoresistive transducers are state-of-the-art monolithic silicon pressure sensors designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 5.0% Maximum Error over 0° to 85°C
- Ideally Suited for Microprocessor or Microcontroller-Based Systems
- Durable Epoxy Unibody and Thermoplastic (PPS) Surface Mount Package
- Temperature Compensated over -40° to +125°C
- Patented Silicon Shear Stress Strain Gauge
- Available in Differential and Gauge Configurations
- Available in Surface Mount (SMT) or Through-hole (DIP) Configurations

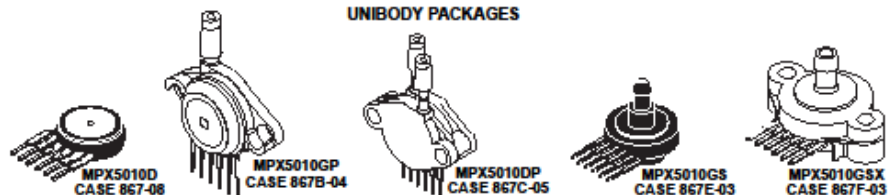
Application Examples

- Hospital Beds
- HVAC
- Respiratory Systems
- Process Control

ORDERING INFORMATION

Device Type	Options	Case No.	MPX Series Order No.	Packing Options	Device Marking
SMALL OUTLINE PACKAGE (MPXV5010G SERIES)					
Basic Elements	Gauge, Element Only, SMT	482	MPXV5010G6U	Reels	MPXV5010G
	Gauge, Element Only, DIP	482B	MPXV5010G7U	Reels	MPXV5010G
Ported Elements	Gauge, Axial Port, SMT	482A	MPXV5010G6U	Reels	MPXV5010G
	Gauge, Axial Port, DIP	482C	MPXV5010G6U	Reels	MPXV5010G
	Gauge, Axial Port, SMT	482A	MPXV5010G6T1	Tape & Reel	MPXV5010G
	Gauge, Side Port, SMT	1369	MPXV5010GP	Trays	MPXV5010G
	Gauge, Dual Port, SMT	1351	MPXV5010DP	Trays	MPXV5010G
UNIBODY PACKAGE (MPX2202 SERIES)					
Basic Element	Differential	867	MPX5010D	—	MPXV5010D
Ported Elements	Differential, Gauge	867C	MPX5010DP	—	MPXV5010DP
	Gauge	867B	MPX5010GP	—	MPXV5010GP
	Gauge, Axial	867E	MPX5010GS	—	MPXV5010D
	Gauge, Axial PC Mount	867F	MPX5010GSX	—	MPXV5010D

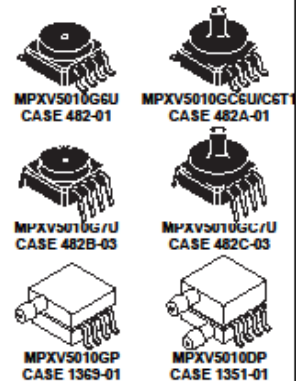
UNIBODY PACKAGES



MPX5010 MPXV5010G SERIES

**INTEGRATED
PRESSURE SENSOR**
0 to 10 kPa (0 to 1.45 psi)
0.2 to 4.7 V OUTPUT

SMALL OUTLINE PACKAGE



UNIBODY PACKAGE PIN NUMBERS⁽¹⁾

1	V _{out}	4	N/C
2	Gnd	5	N/C
3	V _s	6	N/C

1. Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the lead.

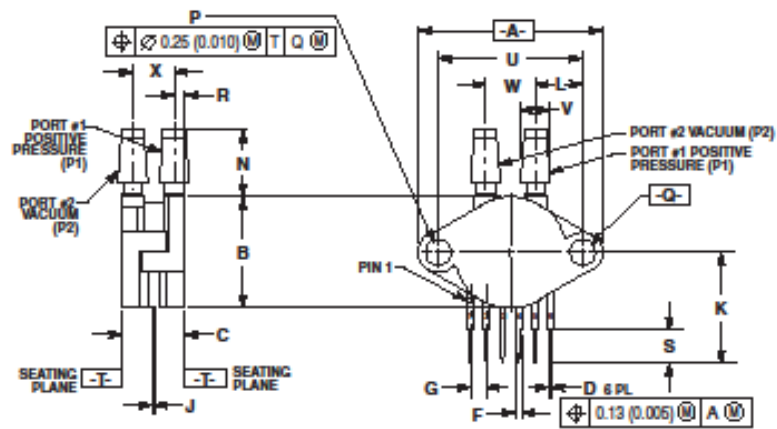
SMALL OUTLINE PACKAGE PIN NUMBERS⁽¹⁾

1	N/C	5	N/C
2	V _s	6	N/C
3	Gnd	7	N/C
4	V _{out}	8	N/C

1. Pins 1, 5, 6, 7, and 8 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the lead.

PACKAGE DIMENSIONS

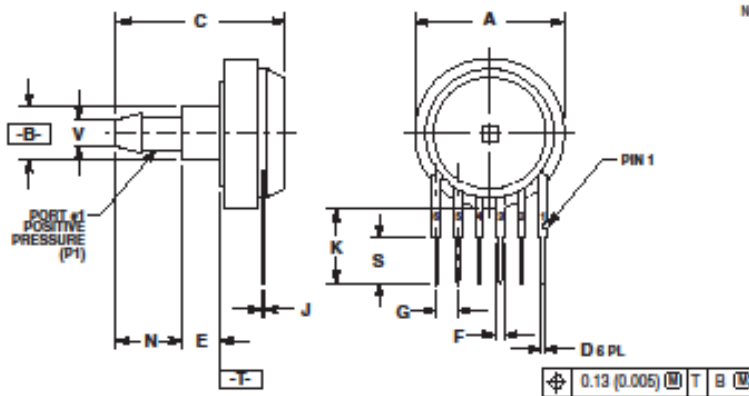
- NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: INCH.



DIM	MIN	MAX	MIN	MAX
A	1.145	1.175	29.28	29.82
B	0.685	0.715	17.40	18.16
C	0.405	0.435	10.29	11.05
D	0.027	0.030	0.68	0.76
F	0.048	0.054	1.22	1.33
G	0.100 BSC	2.54 BSC		
J	0.014	0.016	0.36	0.41
K	0.085	0.125	2.16	3.18
L	0.250	0.300	6.35	7.62
N	0.420	0.440	10.67	11.18
P	0.153	0.159	3.89	4.04
Q	0.153	0.159	3.89	4.04
R	0.083	0.083	2.11	2.11
S	0.229	0.240	5.81	6.10
U	0.010 BSC	25.11 BSC		
V	0.052	0.104	1.32	2.64
W	0.310	0.330	7.87	8.38
X	0.248	0.278	6.30	7.06

- STYLE 1:
PIN 1: V_{cc}
2: GROUND
3: V_{cc}
4: V₁
5: V₂
6: V_{cc}

CASE 867C-05
ISSUE F
UNIBODY PACKAGE



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

DIM	MIN	MAX	MIN	MAX
A	0.680	0.720	17.28	18.28
B	0.245	0.255	6.22	6.48
C	0.780	0.820	19.81	20.82
D	0.027	0.030	0.68	0.76
E	0.178	0.186	4.52	4.72
F	0.048	0.054	1.22	1.33
G	0.100 BSC	2.54 BSC		
J	0.014	0.016	0.36	0.41
K	0.045	0.075	1.14	1.91
N	0.300	0.310	7.62	7.87
S	0.229	0.240	5.81	6.10
V	0.182	0.194	4.62	4.93

- STYLE 1:
PIN 1: V_{cc}
2: GROUND
3: V_{cc}
4: V₁
5: V₂
6: V_{cc}

CASE 867E-03
ISSUE D
UNIBODY PACKAGE

MPX5010

ON-CHIP TEMPERATURE COMPENSATION, CALIBRATION AND SIGNAL CONDITIONING

The performance over temperature is achieved by integrating the shear-stress strain gauge, temperature compensation, calibration and signal conditioning circuitry onto a single monolithic chip.

Figure 2 illustrates the Differential or Gauge configuration in the basic chip carrier (Case 482). A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm.

The MPX5010 and MPXV5010G series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on

sensor performance and long-term reliability. Contact the factory for information regarding media compatibility in your application.

Figure 3 shows the recommended decoupling circuit for interfacing the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

Figure 4 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 0° to 85°C using the decoupling circuit shown in Figure 3. The output will saturate outside of the specified pressure range.

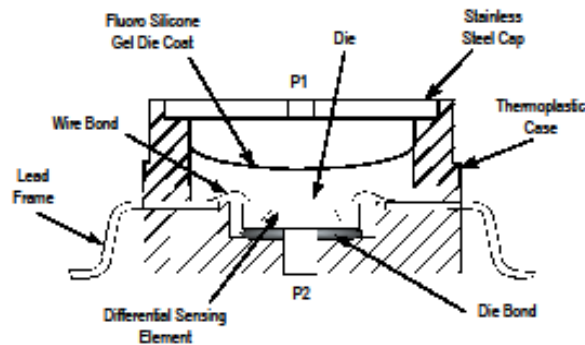


Figure 2. Cross-Sectional Diagram SOP (not to scale)

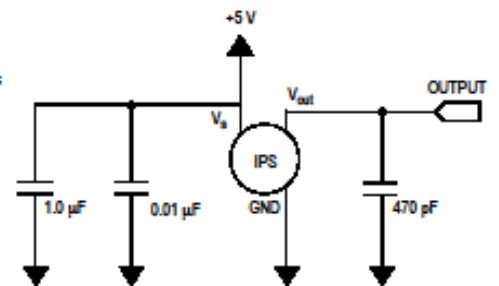


Figure 3. Recommended Power Supply Decoupling and Output Filtering
(For additional output filtering, please refer to Application Note AN1646.)

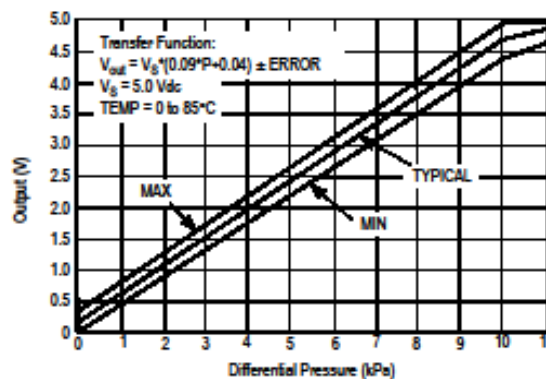


Figure 4. Output versus Pressure Differential

Anexo B3: Motor nema 17

Stepper Motor NEMA 17

This document describes mechanical and electrical specifications for PBC Linear stepper motors; including standard, hollow, and extended shaft variations.

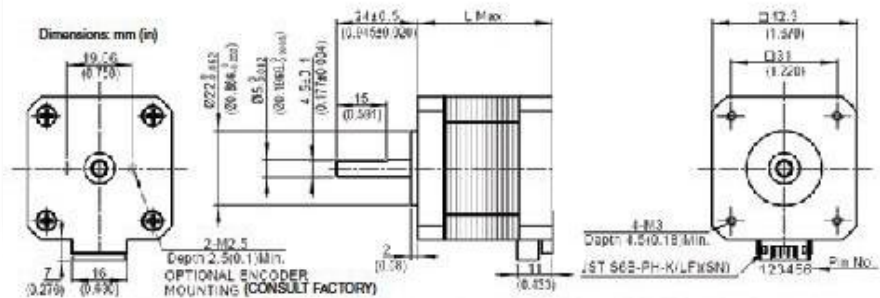


Phases	2
Steps/Revolution	200
Step Accuracy	±5%
Shaft Load	20,000 Hours at 1000 RPM
Axial	25 N (5.6 lbs.) Push 65 N (15 lbs.) Pull
Radial	29 N (6.5 lbs.) At Flat Center
IP Rating	40
Approvals	RoHS
Operating Temp	-20° C to +40° C
Insulation Class	B, 130° C
Insulation Resistance	100 MegOhms

Standard shaft motor shown.

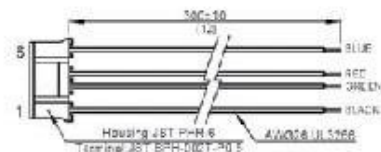
Description	Length	Mounted Rated Current	Mounted Holding Torque	Winding Ohms	Detent Torque	Rotor Inertia	Motor Weight
(Slack)	1" Max	Amps	Nm Typ. oz-in Typ.	±10% @ 20°C Typ.	mNm oz-in	g cm ² oz-in ²	kg lbs
Single	39.8 mm (1.57 in)	2	0.48 68	1.04 2.2	15 2.1	57 0.31	0.28 0.62
Double	48.3 mm (1.90 in)	2	0.63 89	1.3 2.9	25 3.5	82 0.46	0.36 0.79
Triple	62.8 mm (2.47 in)	2	0.83 120	1.49 3.8	30 4.2	123 0.67	0.6 1.3

*All standard motors have plug connector. Consult factory for other options.



Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

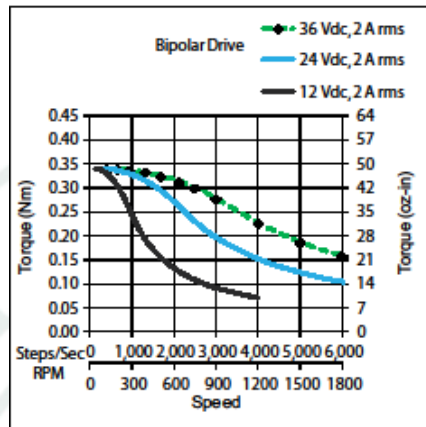
Dimensions: mm (in)
4 Lead Connector, PBC Part#6200490
(Consult factory for optional motor connectors)



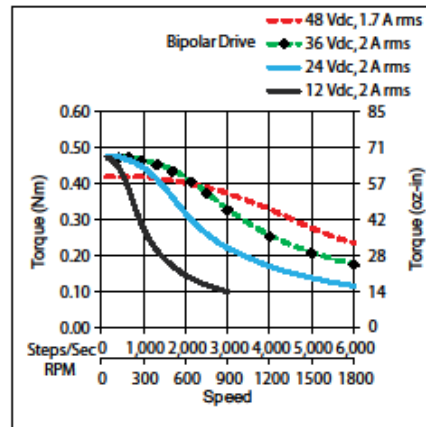
Engineering Your Linear Motion Solutions • pbclinear.com

NEMA 17 Stepper Motor

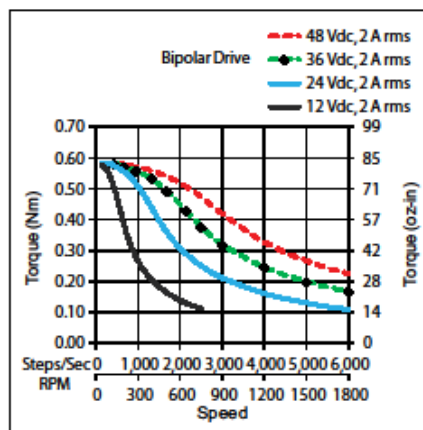
*Performance curves apply to continuous duty cycles.
Consult factory for intermittent cycles or other voltages.



Single Stack



Double Stack



Triple Stack



Engineering Your Linear Motion Solutions • pbclinear.com

Anexo B4: Tornillo Sin fin CNC



Descripciones:

Mano de obra exquisita que garantiza su alto rendimiento.

Producción fina y práctica.

Adecuado para el carril de guía de conducción del motor paso a paso, herramienta de máquina y otros equipos.

Especificaciones:

Material: acero inoxidable

Diámetro del tornillo de plomo: 8mm

Paso del tornillo: 2mm

Plomo de rosca: 2mm

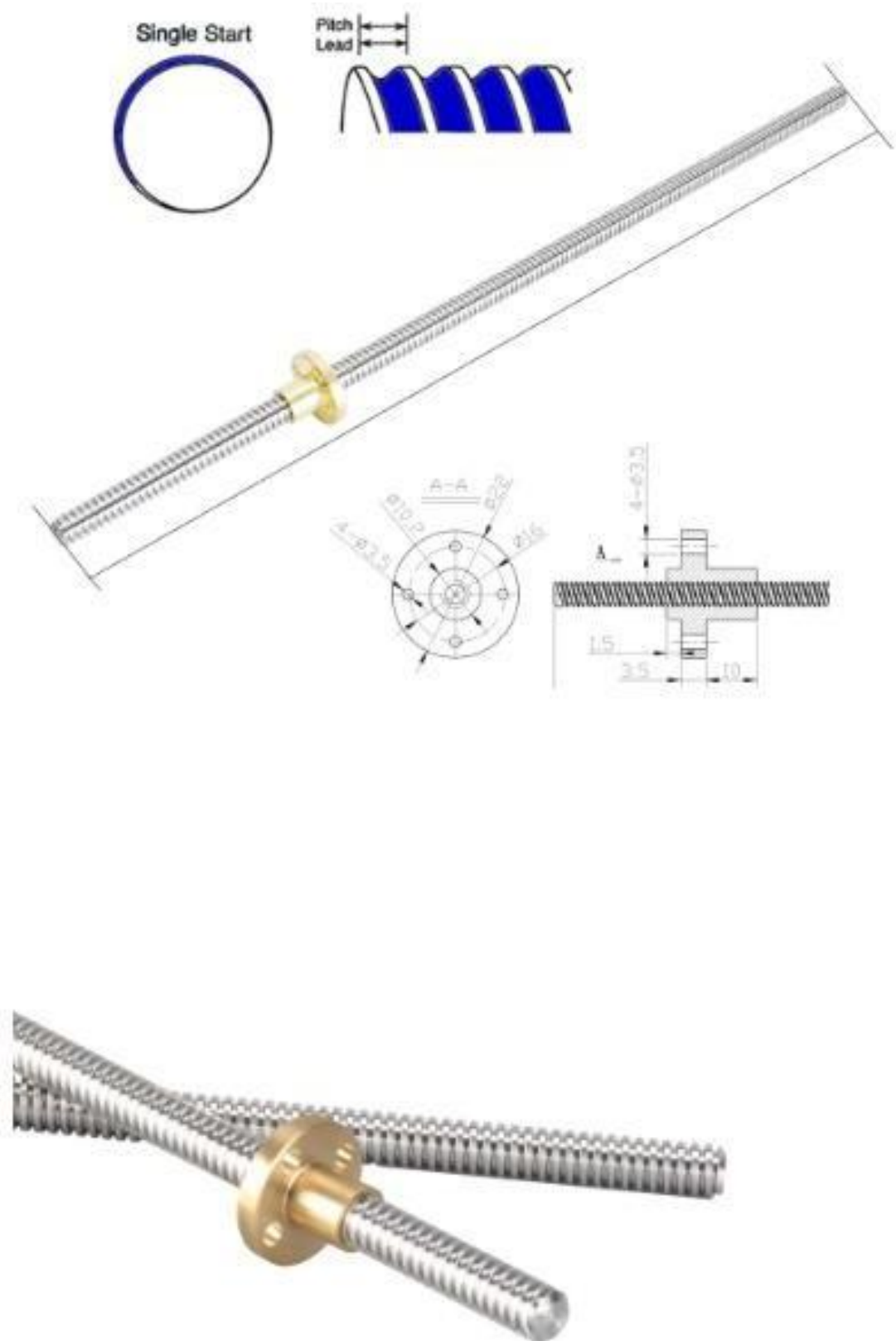
Longitud del tornillo de plomo: 100mm,150mm,200mm,250mm,300mm,330mm,350mm, 400mm,500mm, mm.

Color: Color plateado

Lista de embalaje:

2 piezas x juego de tornillos de plomo con tuercas

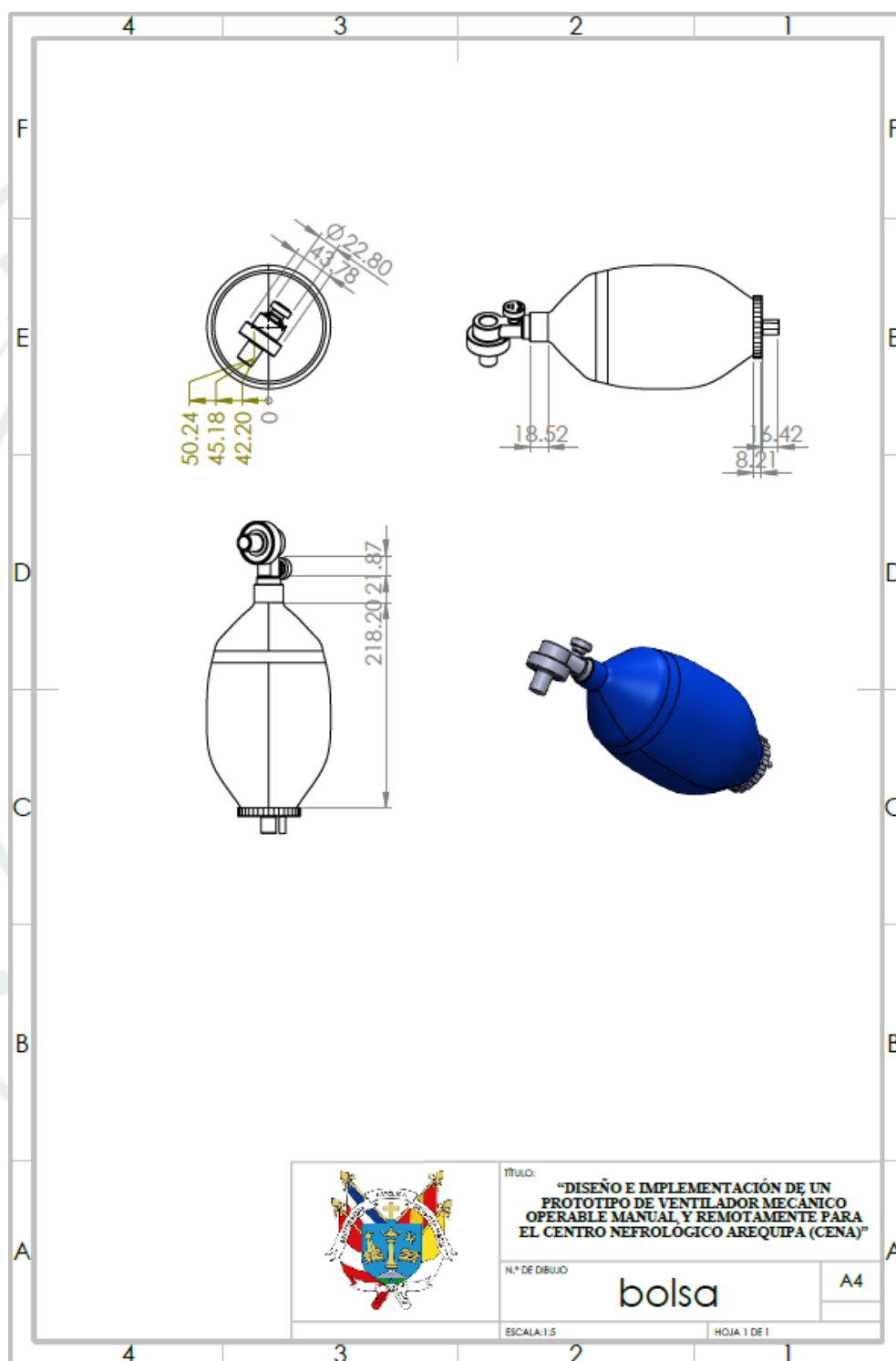




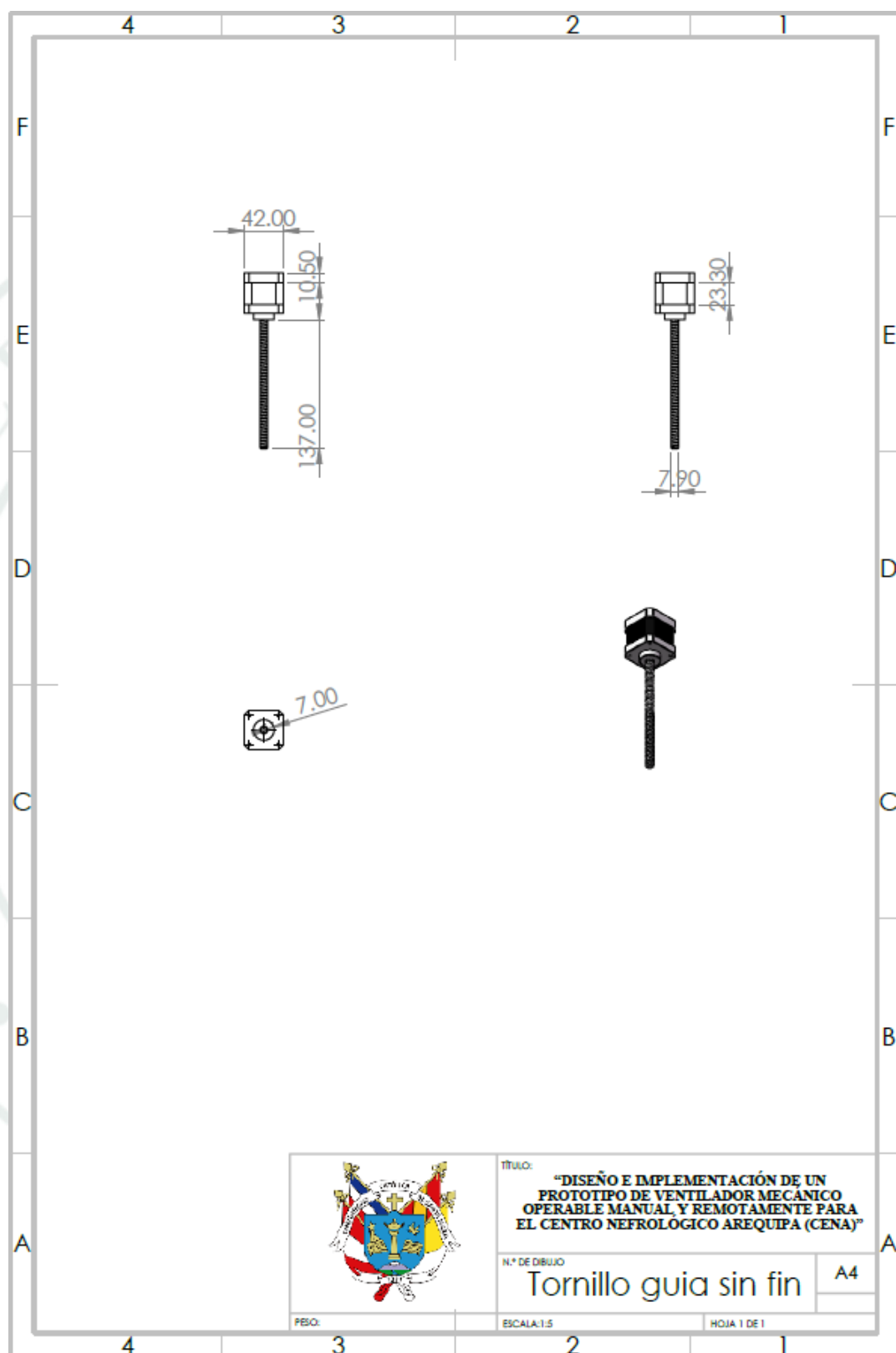
Anexo C: Diseño de componentes y piezas.

En este apartado se muestran los componentes diseñados en SolidWorks.

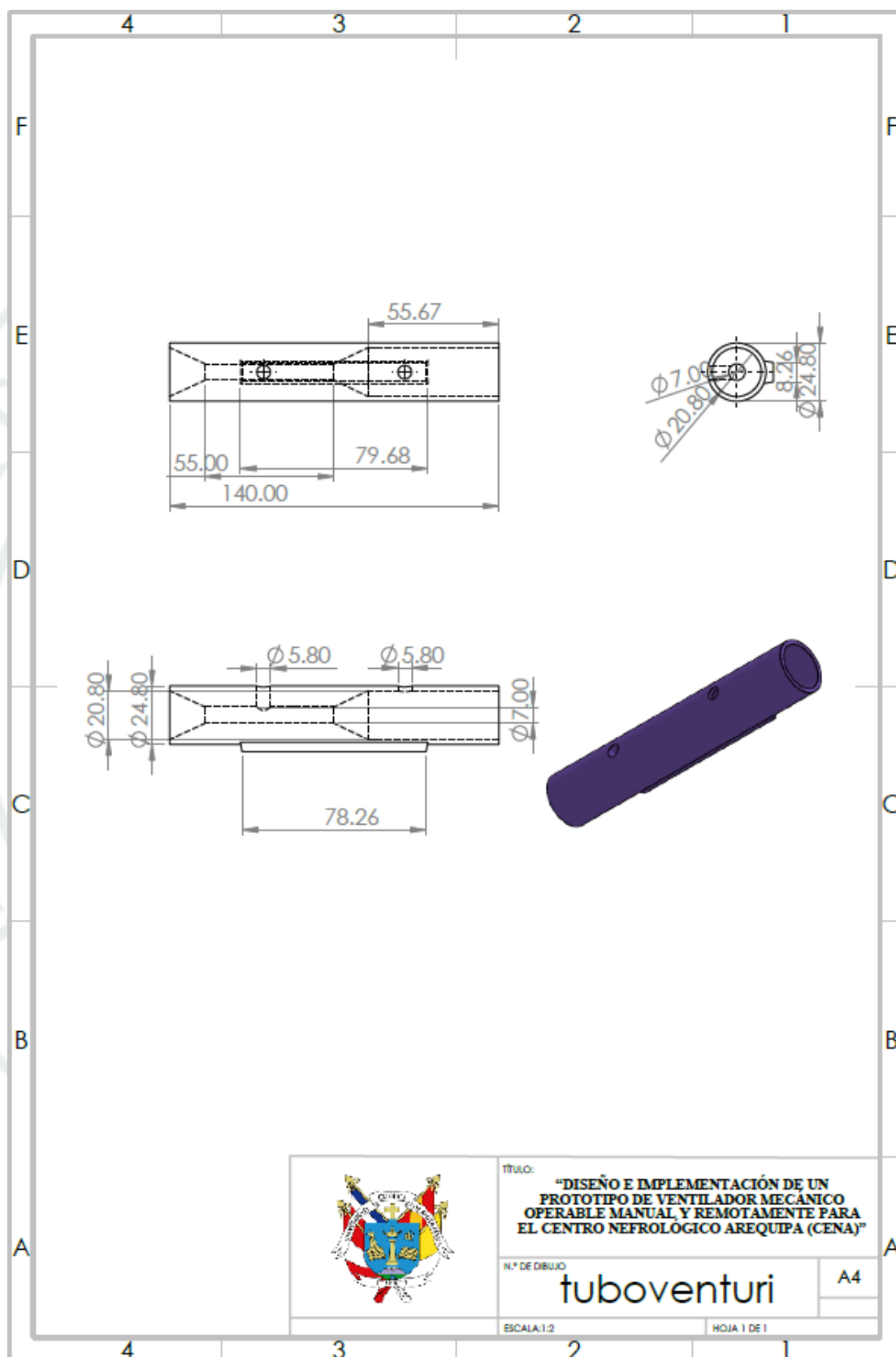
Anexo C1: Bolsa de Ambu



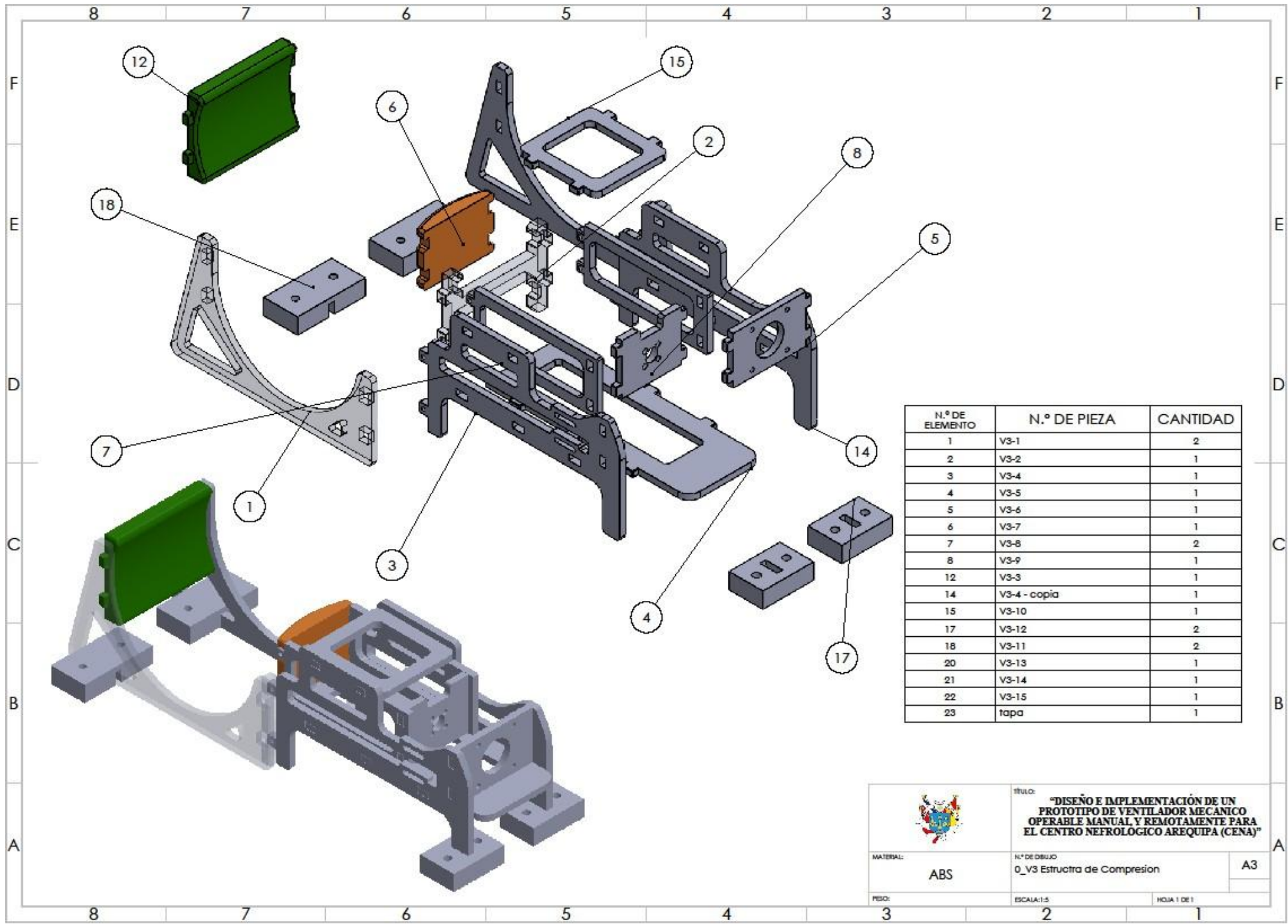
Anexo C2: Motor Mena 17 y tornillo de guía sin fin



Anexo C3: Tubo de Venturi

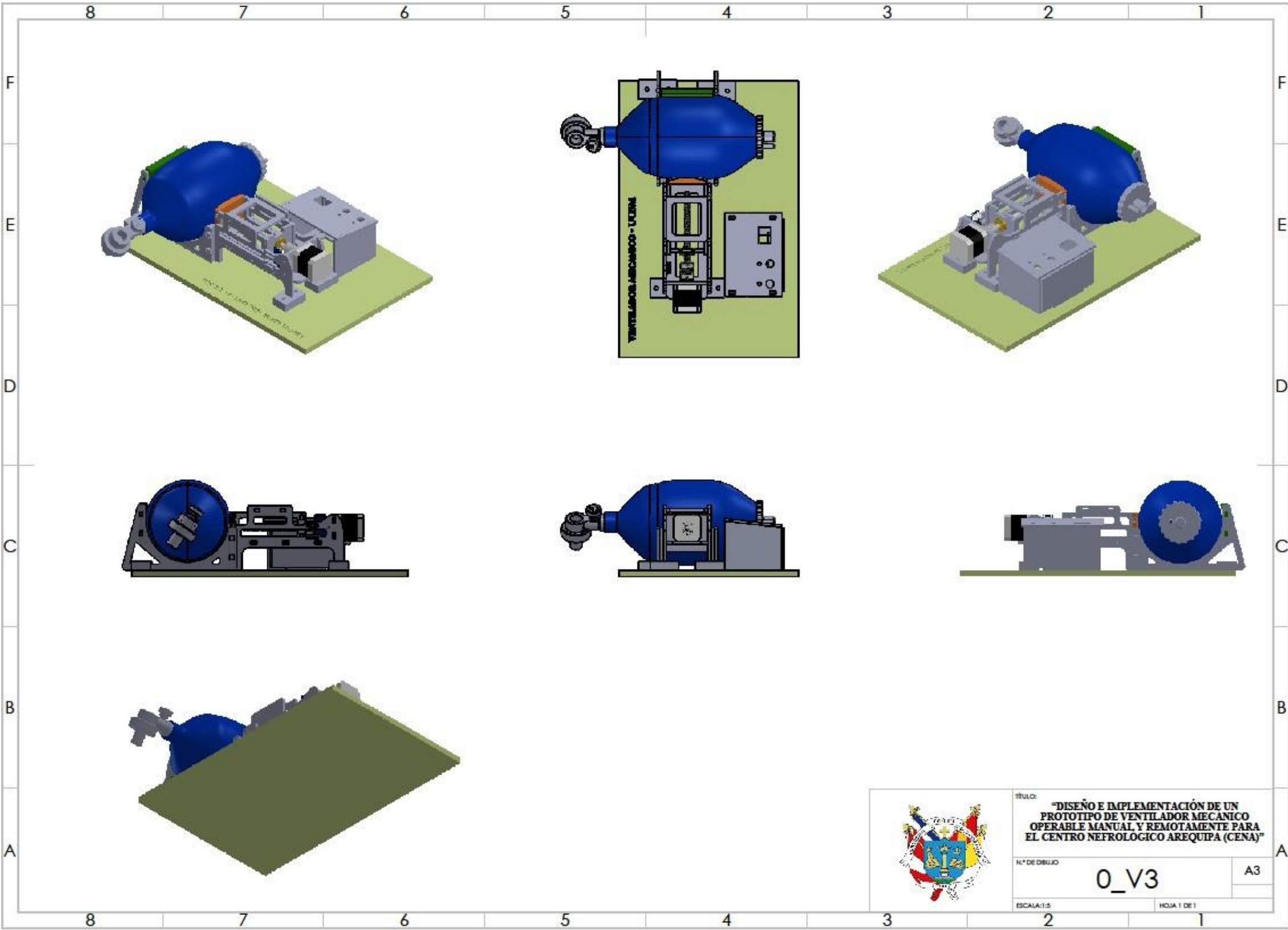


Anexo C4: Estructura de compresión explosionada

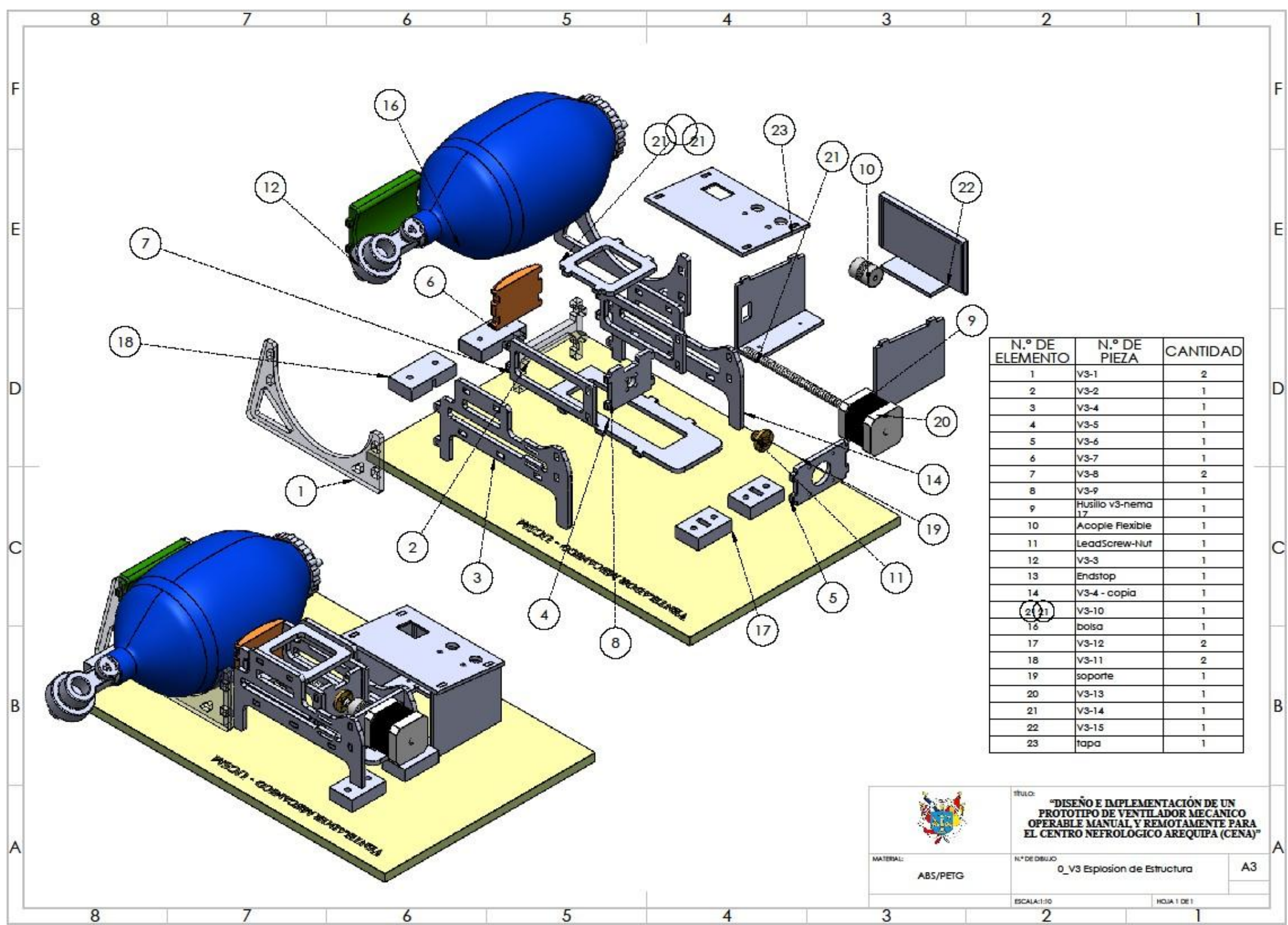


Anexo D: Estructura del Prototipo

Anexo D1: Vistas en el plano ISO_E



Anexo D2: Vista explosiva de la estructura

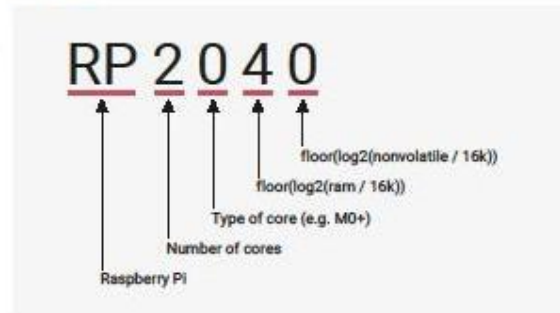


Anexo E: Raspberry Pi Pico R3

1. Number of processor cores (2)
2. Loosely which type of processor (M0+)
3. $\text{floor}(\log_2(\text{ram} / 16\text{k}))$
4. $\text{floor}(\log_2(\text{nonvolatile} / 16\text{k}))$ or 0 if no onboard nonvolatile storage

see Figure 1.

Figure 1. An explanation for the name of the RP2040 chip.



1.2. Summary

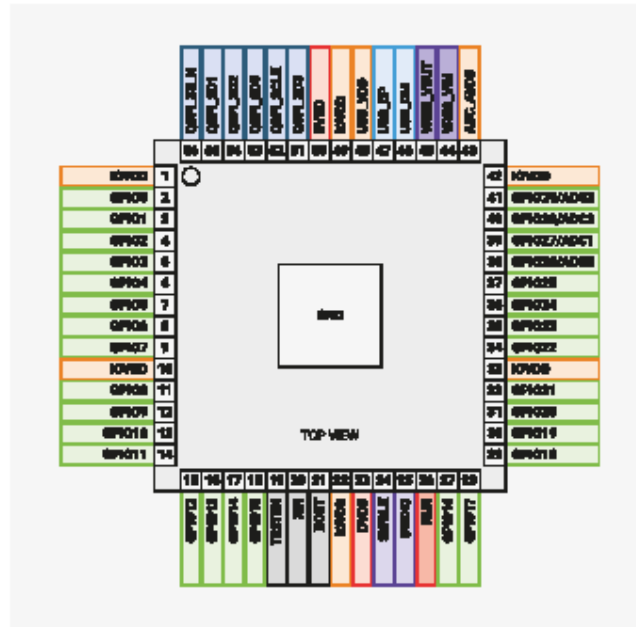
RP2040 is a low-cost, high-performance microcontroller device with flexible digital interfaces. Key features:

- Dual Cortex M0+ processor cores, up to 133MHz
- 264kB of embedded SRAM in 6 banks
- 30 multifunction GPIO
- 6 dedicated IO for SPI Flash (supporting XIP)
- Dedicated hardware for commonly used peripherals
- Programmable IO for extended peripheral support
- 4 channel ADC with internal temperature sensor, 500ksps, 12-bit conversion
- USB 1.1 Host/Device

1.3. The Chip

RP2040 has a dual M0+ processor cores, DMA, internal memory and peripheral blocks connected via AHB/APS bus fabric.

Figure 3. RP2040
Pinout for QFN-56
7x7mm (reduced offset
size)



1.4.2. Pin Descriptions

Table 1. The function
of each pin is briefly
described here. Full
electrical
specifications can be
found in [Chapter 5](#).

Name	Description
GPIOx	General-purpose digital input and output. RP2040 can connect one of a number of internal peripherals to each GPIO, or control GPIOs directly from software.
GPIOx/ADCy	General-purpose digital input and output, with analogue-to-digital converter function. The RP2040 ADC has an analogue multiplexer which can select any one of these pins, and sample the voltage.
QSPIx	Interface to a SPI, Dual-SPI or Quad-SPI flash device, with execute-in-place support. These pins can also be used as software-controlled GPIOs, if they are not required for flash access.
USB_DM and USB_DP	USB controller, supporting Full Speed device and Full/Low Speed host. A 27Ω series termination resistor is required on each pin, but bus pullups and pulldowns are provided internally.
XIN and XOUT	Connect a crystal to RP2040's crystal oscillator. XIN can also be used as a single-ended CMOS clock input, with XOUT disconnected. The USB bootloader requires a 12MHz crystal or 12MHz clock input.
RUN	Global asynchronous reset pin. Reset when driven low, run when driven high. If no external reset is required, this pin can be tied directly to IOVDD.
SWCLK and SWDIO	Access to the Internal Serial Wire Debug multi-drop bus. Provides debug access to both processors, and can be used to download code.
TESTEN	Factory test mode pin. Tie to GND.
GND	Single external ground connection, bonded to a number of internal ground pads on the RP2040 die.
IOVDD	Power supply for digital GPIOs, nominal voltage 1.8V to 3.3V

Name	Description
USB_VDD	Power supply for internal USB Full Speed PHY, nominal voltage 3.3V
ADC_AVDD	Power supply for analogue-to-digital converter, nominal voltage 3.3V
VREG_VIN	Power input for the internal core voltage regulator, nominal voltage 1.8V to 3.3V
VREG_VOUT	Power output for the internal core voltage regulator, nominal voltage 1.1V, 100mA max current
DVDD	Digital core power supply, nominal voltage 1.1V. Can be connected to VREG_VOUT, or to some other board-level power supply.

1.4.3. GPIO Functions

Each individual GPIO pin can be connected to an internal peripheral via the GPIO functions defined below. Some internal peripheral connections appear in multiple places to allow some system level flexibility. SIO, PIO0 and PIO1 can connect to all GPIO pins and are controlled by software (or software controlled state machines) so can be used to implement many functions.

Table 2. General Purpose Input/Output (GPIO) Bank 0 Functions

GPIO	Function								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	SPI0 RX	UART0 TX	I2C0 SDA	PWM0 A	SIO	PIO0	PIO1		USB OVCUR DET
1	SPI0 CSn	UART0 RX	I2C0 SCL	PWM0 B	SIO	PIO0	PIO1		USB VBUS DET
2	SPI0 SCK	UART0 CTS	I2C1 SDA	PWM1 A	SIO	PIO0	PIO1		USB VBUS EN
3	SPI0 TX	UART0 RTS	I2C1 SCL	PWM1 B	SIO	PIO0	PIO1		USB OVCUR DET
4	SPI0 RX	UART1 TX	I2C0 SDA	PWM2 A	SIO	PIO0	PIO1		USB VBUS DET
5	SPI0 CSn	UART1 RX	I2C0 SCL	PWM2 B	SIO	PIO0	PIO1		USB VBUS EN
6	SPI0 SCK	UART1 CTS	I2C1 SDA	PWM3 A	SIO	PIO0	PIO1		USB OVCUR DET
7	SPI0 TX	UART1 RTS	I2C1 SCL	PWM3 B	SIO	PIO0	PIO1		USB VBUS DET
8	SPI1 RX	UART1 TX	I2C0 SDA	PWM4 A	SIO	PIO0	PIO1		USB VBUS EN
9	SPI1 CSn	UART1 RX	I2C0 SCL	PWM4 B	SIO	PIO0	PIO1		USB OVCUR DET
10	SPI1 SCK	UART1 CTS	I2C1 SDA	PWM5 A	SIO	PIO0	PIO1		USB VBUS DET
11	SPI1 TX	UART1 RTS	I2C1 SCL	PWM5 B	SIO	PIO0	PIO1		USB VBUS EN
12	SPI1 RX	UART0 TX	I2C0 SDA	PWM6 A	SIO	PIO0	PIO1		USB OVCUR DET
13	SPI1 CSn	UART0 RX	I2C0 SCL	PWM6 B	SIO	PIO0	PIO1		USB VBUS DET
14	SPI1 SCK	UART0 CTS	I2C1 SDA	PWM7 A	SIO	PIO0	PIO1		USB VBUS EN
15	SPI1 TX	UART0 RTS	I2C1 SCL	PWM7 B	SIO	PIO0	PIO1		USB OVCUR DET
16	SPI0 RX	UART0 TX	I2C0 SDA	PWM0 A	SIO	PIO0	PIO1		USB VBUS DET
17	SPI0 CSn	UART0 RX	I2C0 SCL	PWM0 B	SIO	PIO0	PIO1		USB VBUS EN
18	SPI0 SCK	UART0 CTS	I2C1 SDA	PWM1 A	SIO	PIO0	PIO1		USB OVCUR DET
19	SPI0 TX	UART0 RTS	I2C1 SCL	PWM1 B	SIO	PIO0	PIO1		USB VBUS DET
20	SPI0 RX	UART1 TX	I2C0 SDA	PWM2 A	SIO	PIO0	PIO1	CLOCK GPIO0	USB VBUS EN
21	SPI0 CSn	UART1 RX	I2C0 SCL	PWM2 B	SIO	PIO0	PIO1	CLOCK GP0UT0	USB OVCUR DET

	Function								
22	SPI0 SCK	UART1 CTS	I2C1 SDA	PWM3 A	SIO	PI00	PI01	CLOCK GPIN1	USB VBUS DET
23	SPI0 TX	UART1 RTS	I2C1 SCL	PWM3 B	SIO	PI00	PI01	CLOCK GPOUT1	USB VBUS EN
24	SPI1 RX	UART1 TX	I2C0 SDA	PWM4 A	SIO	PI00	PI01	CLOCK GPOUT2	USB OVCUR DET
25	SPI1 CSn	UART1 RX	I2C0 SCL	PWM4 B	SIO	PI00	PI01	CLOCK GPOUT3	USB VBUS DET
26	SPI1 SCK	UART1 CTS	I2C1 SDA	PWM5 A	SIO	PI00	PI01		USB VBUS EN
27	SPI1 TX	UART1 RTS	I2C1 SCL	PWM5 B	SIO	PI00	PI01		USB OVCUR DET
28	SPI1 RX	UART0 TX	I2C0 SDA	PWM6 A	SIO	PI00	PI01		USB VBUS DET
29	SPI1 CSn	UART0 RX	I2C0 SCL	PWM6 B	SIO	PI00	PI01		USB VBUS EN

Table 3. GPIO bank 0
function descriptions

Function Name	Description
SPIx	Connect one of the internal PL022 SPI peripherals to GPIO
UARTx	Connect one of the internal PL011 UART peripherals to GPIO
I2Cx	Connect one of the internal DW I2C peripherals to GPIO
PWMx A/B	Connect a PWM slice to GPIO. There are eight PWM slices, each with two output channels (A/B). The B pin can also be used as an input, for frequency and duty cycle measurement.
SIO	Software control of GPIO, from the single-cycle IO (SIO) block. The SIO function (<i>f5</i>) must be selected for the processors to drive a GPIO, but the input is always connected, so software can check the state of GPIOs at any time.
PIOx	Connect one of the programmable IO blocks (PIO) to GPIO. PIO can implement a wide variety of interfaces, and has its own internal pin mapping hardware, allowing flexible placement of digital interfaces on bank 0 GPIOs. The PIO function (<i>f6, f7</i>) must be selected for PIO to drive a GPIO, but the input is always connected, so the PIOs can always see the state of all pins.
CLOCK GPINx	General purpose clock inputs. Can be routed to a number of internal clock domains on RP2040, e.g. to provide a 1Hz clock for the RTC, or can be connected to an internal frequency counter.
CLOCK GPOUTx	General purpose clock outputs. Can drive a number of internal clocks (including PLL outputs) onto GPIOs, with optional integer divide.
USB OVCUR DET/VBUS DET/VBUS EN	USB power control signals to/from the internal USB controller

Anexo F: Código

Anexo F1: Micropython R1

A continuación, se muestra el código programado para el microcontrolador de modo local, es decir este es el principal.

```
1  #LIBRERIAS
2  import time
3  import machine
4  from machine import Pin
5  from time import sleep
6
7  #PINES DE ENTRADAS Y SALIDAS
8
9  #PULSADORES
10 boton_infante = Pin(12, Pin.IN, Pin.PULL_UP)
11 boton_adulto = Pin(13, Pin.IN, Pin.PULL_UP)
12
13 boton_infante_digital = Pin(21, Pin.IN, Pin.PULL_UP)
14 boton_adulto_digital = Pin(20, Pin.IN, Pin.PULL_UP)
15
16 #INDICADORES
17 led_rojo = Pin(10, Pin.OUT)
18 led_azul = Pin(11, Pin.OUT)
19 LED = Pin(25, Pin.OUT)
20 #SENSOR DE FIN DE CARRIL
21 carril = machine.Pin(15, machine.Pin.IN, machine.Pin.PULL_UP)
22 #MOTOR DRIVER TB6600
23 #DIR-, PUL-, ENA- a GND. Conecté DIR+ al pin 4, PUL+ al pin 3, ENA+ al pin 2.
24 dirPin = Pin(6, Pin.OUT)
25 stepPin = Pin(3, Pin.OUT)
26 enaPin = Pin(2, Pin.OUT)
27 enaPin.low()
28
29 #PARAMETROS
30 contador_uno = 1
31 modo_uno='L'
32 contador_dos = 1
33 modo_dos='L'
34
35 #FUNCIONES
36 #FUNCIONES DE INTERRUPCION
37
38 def boton_uno_interrupt_digital(Pin):
39     global contador_uno
40     print (contador_uno)
41     contador_uno +=1
42     if contador_uno > 2:
43         contador_uno =1
44     if contador_uno ==2:
45         led_azul.on() #Encender led
46         global modo_uno
47         modo_uno = 'K' # Guardar variable
48     else:
```

```

49         led_azul.off()
50         modo_uno='L' # Apagar led y guardar variable
51     boton_infante_digital.irq(handler = boton_uno_interrupt_digital, trigger=Pin.IRQ_FALLING)
52
53
54     def boton_dos_interrupt_digital(Pin):
55         global contador_dos
56         print (contador_dos)
57         contador_dos +=1
58         if contador_dos > 2:
59             contador_dos =1
60         if contador_dos ==2:
61             led_rojo.on() # Encender led
62             global modo_dos
63             modo_dos = 'N' # Guardar variable
64             #print(modo_dos)
65         else:
66             led_rojo.off()
67             modo_dos='L' # Apagar led y guardar variable
68     boton_adulto_digital.irq(handler = boton_dos_interrupt_digital, trigger=Pin.IRQ_FALLING)
69
70     def boton_uno_interrupt(Pin):
71         global contador_uno
72         print (contador_uno)
73         contador_uno +=1
74         if contador_uno > 2:
75             contador_uno =1
76         if contador_uno ==2:
77             led_azul.on() # Encender led
78             global modo_uno
79             modo_uno = 'K' # Guardar variable
80         else:
81             led_azul.off()
82             modo_uno='L' # Apagar led y guardar variable
83     boton_infante.irq(handler = boton_uno_interrupt, trigger=Pin.IRQ_FALLING)
84     def boton_dos_interrupt(Pin):
85         global contador_dos
86         print (contador_dos)
87         contador_dos +=1
88         if contador_dos > 2:
89             contador_dos =1
90         if contador_dos ==2:
91             led_rojo.on() # Encender led
92             global modo_dos
93             modo_dos = 'N' # Guardar variable
94             #print(modo_dos)
95         else:
96             led_rojo.off()
97             modo_dos='L' # Apagar led y guardar variable
98     boton_adulto.irq(handler = boton_dos_interrupt, trigger=Pin.IRQ_FALLING)
99
100     #FUNCIONES DE MODO DE CONTROL DEL MOTOR DE PASO
101
102     def modo_adulto():
103         if first==0:
104             LED.on()
105             dirPin.high()
106             for x in range (1500):
107                 #print('Izquierda')
108                 stepPin.low()

```

```

109         time.sleep_us(600)
110         stepPin.high()
111         time.sleep_us(600)
112         if modo_uno == 'L':
113             break
114     else:
115         LED.off()
116         # print('Derecha')
117         dirPin.low()
118         stepPin.low()
119         time.sleep_us(600)
120         stepPin.high()
121         time.sleep_us(600)
122
123
124
125     def modo_infante():
126         if first==0:
127             LED.on()
128             dirPin.high()
129             for x in range (1300): # Rango de alcance
130                 #print('Izquierda')
131                 stepPin.low()
132                 time.sleep_us(600) #Induccion de corriente para la Velocidad
133                 stepPin.high()
134                 time.sleep_us(600) #Velocidad
135                 if modo_dos == 'L':
136                     break
137             else:
138                 LED.off()
139                 #print('Derecha')
140                 dirPin.low()
141                 stepPin.low()
142                 time.sleep_us(600)
143                 stepPin.high()
144                 time.sleep_us(600)
145
146
147
148     #BUCLE PRINCIPAL
149
150     while True:
151         if modo_uno == 'K' and modo_dos == 'L': #Led rojo encendido y led azul apagado
152             first = carril.value() #llamar a carril con valor 1 o 0
153             modo_adulto() #Llamada de funcion
154             #print(modo_uno)
155             print('Modo Adulto')
156
157         if modo_dos == 'N' and modo_uno == 'L': #Led rojo apagado y led azul encendido
158             first = carril.value()
159             modo_infante()
160             #print(modo_dos)
161             print('Modo Infante')
162         if (modo_dos == 'L' and modo_uno == 'L') or (modo_dos == 'N' and modo_uno == 'N'):
163             sleep(1)
164             #print(modo_dos)
165             print('Modo Neutro')
166
167

```

Anexo F2: Micropython R2

```
1  #LIBRERIAS
2  from machine import UART, Pin
3  from machine import Timer
4  from machine import ADC
5  import time
6  import machine
7  from math import sqrt
8  from machine import Pin, SPI
9  from ssd1306 import SSD1306_SPI
10 from time import sleep
11 import utime
12 from machine import I2C, Pin
13
14 #VARIABLES Y PARAMETROS
15 dev= I2C(1, scl=Pin(15), sda=Pin(14))
16 print(dev.scan())
17 address=72
18
19 ANCHO = 128
20 ALTO = 64
21 spi = SPI(0, 100000, mosi=Pin(19), sck=Pin(18))
22 oled = SSD1306_SPI(ANCHO, ALTO, spi, Pin(17),Pin(20), Pin(16))
23 #oled = SSD1306_SPI(WIDTH, HEIGHT, spi, dc,rst, cs) use GPIO PIN NUMBERS
24
25 oled.fill(0)
26 oled.text("Ventilador",22,0)
27 oled.text("Mecanico",30,16)
28 oled.text("UCSM",42,32)
29 #oled.text(".. Inicio ..",10,52)
30 oled.show()
31 #indicador_azul = Pin(15, Pin.IN)
32 #indicador_rojo = Pin(16, Pin.IN)
33
```

```

34 #FUNCIONES DE LECTURA Y CONVERSION PARA EL MODULO ADS1115
35 def readConfig():
36     dev.writeto(address, bytearray([1]))
37     result = dev.readfrom(address,2)
38     return result[0]<<8 | result[0]
39 def readValueFrom(channel):
40     config =readConfig()
41     config &= ~(7<<12) #clear MUX bits
42     config &= ~(7<<9) #clear PGA
43     config |= (7 & (4+channel))<<12
44     config |= (1<<15)#trigger next convert
45     config |= (1<<9)#gain 4.096 volts
46     config = [int(config>>i & 0xff) for i in[8,0]]
47     dev.writeto(address, bytearray ([1] + config))
48     config = readConfig()
49     while (config & 0x8000) == 0:
50         config = readConfig()
51     dev.writeto(address, bytearray([0]))
52     result =dev.readfrom(address, 2)
53     return result[0]<<8 | result[0]
54 def readVotlsFrom(channel):
55     value=readValueFrom(channel)
56     return ((4.096*2)/0xffff)*value
57
58 val=[0,0,0,0]
59
60 def EnviarLectura(t): #El método debe tener un argumento en este caso t
61     print ('Enviando...')
62     val[0] = readVotlsFrom(0)
63     val[1] = readValueFrom(1)
64     val[2] = readValueFrom(2)
65     val[3] = readValueFrom(3)
66     print(val[0])
67     Voltaje=val[0]
68     #ValorADC=adc.read_u16() #Leer valor de #Para resistencia de divisor de voltaje
69     #Voltaje=ValorADC*(3.3/65535) #Conversion a voltaje 16 bits #Para resistencia
70     #Presion=((Voltaje-0.04*5-0.01)/(0.09*5))*100#Formula de DataSheet MPX5010DP KPa
71     Presion=((Voltaje-0.04*3.3+0.04)/(0.09*3.3))*10.1978#Formula de DataSheet MPX5010D
72     # Presion= Vout-0.04*Vs +- Tol / 0.09Vs
73     # Vs= Voltaje de alimentación
74     AP=Presion/10.1978 #kPa
75     # 1kPa=1000*Kg*m^-1*s^-2
76     # Densidad de agua p=1.225 kg*m^-3
77     # Velocidad por el tubo de conducto
78     #
79     #..... / 2*AP*1000 kg*m^-1*s^-2
80     #..... V = / ----- =m/s
81     #..... √ p*(76.96) kg*m^-3
82     #.....
83
84     V= sqrt((2*AP*1000)/(1.225*76.96))
85
86     # Dieametro del tubo 20.8 mm
87     # pi*diametro^2

```

```

88 # A =----- =Area m^2
89 #      4
90 Area=3.397946*10**(-4) #
91 # 1 m^3= 1000L
92 # Caudal=Q=V*A m^3/s
93 Q=Area*V
94 QLitros= Q*1000*60
95 print(QLitros)
96 print (Presion)
97 oled.fill(0)
98 oled.text("Ventilador",22,0)
99 oled.text("Mecanico",30,16)
100 oled.text("UCSM",42,32)
101 oled.text(" P = " + str(round(Presion,4)),10,52)
102 #oled.text(".. Inicio ..",10,52)
103 oled.show()
104 uart0.write(str(round(Presion,2))) #Enviar valor convertido del ADC
105 uart0.write("\n") #Escribir el carácter terminador de la cadena a enviar
106 uart1.write(str(round(Presion,2))) #Enviar valor convertido del ADC
107 uart1.write("\n") #Escribir el carácter terminador de la cadena a enviar
108 #if indicador_azul.value()==0:
109 #    oled.fill_rect(0,0,120,42,0)
110 #    oled.text("MODO ADULTO",0,0)
111 #    oled.text("Contando: ",10,32)
112 #    oled.text("UCSM",42,32)
113 #    oled.show()
114 #elif indicador_rojo.value()==0 :
115 #    oled.fill_rect()
116 #    oled.text("MODO INFANTE",0,0)
117 #    oled.text("Contando: ",10,32)
118 #    oled.text("UCSM",42,32)
119 #    oled.show()
120 #adc=machine.ADC(0) #ADC(Pin(28)) #Canal Analogo en el pin 28. #para resistencia
121 uart1= UART(1,baudrate=9600, tx=Pin(4), rx=Pin(5)) #Configurar UART
122 uart0= UART(0,baudrate=9600, tx=Pin(0), rx=Pin(1)) #Configurar UART
123 tim = Timer(period=50, mode=Timer.PERIODIC, callback=EnviarLectura) #Crear Timer
124 #con callback cada 1.5seg
125 led_azul = Pin(10, Pin.OUT)
126 led_rojo = Pin(11, Pin.OUT)
127 # BUCLE PRINCIPAL DE CONTROL
128 while True:
129
130     if uart1.any()>0: #Si hay datos disponibles en puerto serial
131         print ('detectado') #imprimir detectado
132         valor=uart1.read() #Leer dato enviado desde consola
133         valorsrt=str(valor.decode('UTF-8')) #Transformar a string mediante
134         #decodificador UTF-8
135         print (valorsrt) #Imprimir valor leído
136         #Mover el motor paso a paso de acuerdo al comando
137         if (valorsrt=='D'):
138             led_azul.on()
139             led_azul.off()
140             print('modo adulto')
141         elif (valorsrt=='T'):

```

```
142 led_rojo.on()
143 led_rojo.off()
144 print('modo infante')
145
146
147 if uart0.any()>0: #Si hay datos disponibles en puerto serial
148     print('detectado') #imprimir detectado
149     valor=uart0.read() #Leer dato enviado desde consola
150     valorsrt=str(valor.decode('UTF-8')) #Transformar a string
151     #mediante decodificador UTF-8
152     print(valorsrt) #Imprimir valor leído
153
154
155 if "D" in valorsrt:
156     led_azul.on()
157     led_azul.off()
158     print('modo adulto')
159
160 if "I" in valorsrt:
161     led_rojo.on()
162     led_rojo.off()
163     print('modo infante')
```

Anexo F3: Visual Basic (GUI)

A continuación, se muestra el código desarrollado en el software de Visual Basic 2019.

```
1 Imports System.Linq
2 Imports System.Data.SqlClient
3 Imports System.Data.OleDb
4 Imports Microsoft.Office.Core
5 Imports Excel = Microsoft.Office.Interop.Excel
6 Imports ExcelAutoFormat = Microsoft.Office.Interop.Excel.XlRangeAutoFormat
7 Imports Microsoft.Office.Interop
8 Imports System.IO
9 Imports System.Xml.XPath
10 Imports System.Data
11 Imports System.Xml
12 Imports System.Reflection
13 Imports System.Windows.Forms.DataVisualization.Charting
14
15 Public Class Form1
16
17     Dim i As New Integer 'Variable para muestra leida
18     Dim LecturaSerial As String 'Variable para almacenar muestra de puerto serial
19     Dim LecturaSerialf As Integer
20     Dim ChartLimit As Integer = 100
21
22
23
24     Private Sub Chart1_Click(sender As Object, e As EventArgs) Handles Chart1.Click
25
```

```

26 End Sub
27
28 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
29     SerialPort1.PortName = ComboBox1.SelectedItem.ToString 'Obtener nombre de puerto seleccionado del usuario para abrirlo
30     Label2.Text = ComboBox1.SelectedItem.ToString
31     SerialPort1.BaudRate = 9600 '9600 Baudios de velocidad
32     If SerialPort1.IsOpen Then
33         MessageBox.Show("Puerto Abierto, cierre el puerto y vuelva a intentarlo")
34     Else
35         Try
36             SerialPort1.Open() 'Abrir puerto serial
37             MessageBox.Show("Conectado...") 'Mostrar mensaje
38             Timer1.Interval = 10 'Ajustar Timer a intervalo de 1 segundo, el timer leera los datos del buffer del puerto serial
39             AddHandler Timer1.Tick, AddressOf ActualizarGrafica 'Generar Handler, cada vez que pase 1 segundo leer el buffer serial
40             Timer1.Start() 'Iniciar Timer
41
42             'AddHandler SerialPort1.DataReceived, AddressOf ActualizarGrafica 'Si hay datos de recepción, generar interrupcion (IRQ)
43         Catch
44             MessageBox.Show("Puerto inexistente, chequee el puerto de comunicación...")
45         End Try
46
47     End If
48 End Sub
49
50 Public Sub ActualizarGrafica(sender As Object, e As EventArgs)
51
52     If SerialPort1.IsOpen Then
53         If SerialPort1.BytesToRead > 0 Then 'Si hay Bytes disponibles en el Buffer
54             LecturaSerial = SerialPort1.ReadLine 'Leer hasta que hay un terminador de la cadena
55             Label2.Text = "P: " + LecturaSerial + " cmH2O" 'Asignar a etiqueta 2 el valor leído

```

```

56
57 Dim Presion = LecturaSerial.Length
58 Dim PcmH2O = Presion / 10.1978
59 Dim V = ((2 * PcmH2O * 1000) / (1.225 * 76.96)) ^ (1 / 2)
60
61 V = sqrt((2 * AP * 1000) / (1.225 * 76.96))
62 '# Dieametro del tubo 20.8 mm
63 '# pi*diametro^2
64 '# A = ----- =Area m^2
65 '#      4
66 '# Area = 3.397946 * 10 ** (-4) #
67 '# 1 m^3= 1000L
68 '# Caudal=Q=V*A m^3/s
69 '# Q = Area * V
70 '# QLitros = Q * 1000 * 60
71 Dim Area = 0.3397 * 10 ^ (-3)
72 Dim QLitros = Area * V * 1000 * 60
73 '#Escalamos
74 Dim Qabsoluto = (QLitros - 58)
75
76 Dim Vabsoluto = Math.Round((Qabsoluto / 60), 5)
77 Dim resultado As String = Vabsoluto
78 Label3.Text = "V: " + resultado + " Litros" '#Asignar a etiqueta 2 el valor leído
79
80 '#i = i + 1 '#Muestra actual
81 Chart1.Series("Presion").Points.AddXY(DateTime.Now.ToLongTimeString, LecturaSerial) '#Actualizar el chart con los datos
82 leídos
83 Chart2.Series("Volumen").Points.AddXY(DateTime.Now.ToLongTimeString, Vabsoluto) '#Actualizar el chart con los datos leídos
84 '#DateTime.Now.ToLongTimeString
85 End If

```

```

86     End If
87
88 End Sub
89
90 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
91     If SerialPort1.IsOpen Then
92         SerialPort1.Close() Cerrar puerto serie
93         Timer1.Stop() Detener Timer
94         MessageBox.Show("Desconectado...")
95     Else
96         MessageBox.Show("El puerto seleccionado no esta abierto...")
97     End If
98 End Sub
99
100 Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
101     ObtenerPuertos()
102 End Sub
103 Public Sub ObtenerPuertos()
104     ComboBox1.Items.Clear() Borrar Todos los items del combobox cuando se actualizan los puertos
105
106     For Each Puerto As String In My.Computer.Ports.SerialPortNames Leer puertos disponibles en PC
107         ComboBox1.Items.Add(Puerto)
108     Next
109 End Sub
110
111 Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ComboBox1.SelectedIndexChanged
112
113 End Sub
114
115

```

```

116
117 Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
118     'Comando para adulto
119     SerialPort1.Write("D")
120     'PictureBox2.Visible = True
121 End Sub
122
123 Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
124     'Comando para Infante
125     SerialPort1.Write("I")
126     'PictureBox3.Visible = True
127 End Sub
128
129
130
131 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
132     ObtenerPuertos() 'Actualizar puertos disponibles
133     For i = 0 To 50 Step 1
134         Chart1.Series("Presion").Points.AddY(0)
135         If Chart1.Series(0).Points.Count = ChartLimit Then
136             Chart1.Series(0).Points.RemoveAt(0)
137         End If
138     Next
139     For i = 0 To 50 Step 1
140         Chart2.Series("Volumen").Points.AddY(0)
141         If Chart2.Series(0).Points.Count = ChartLimit Then
142             Chart2.Series(0).Points.RemoveAt(0)
143         End If
144     Next
145     -----

```

```

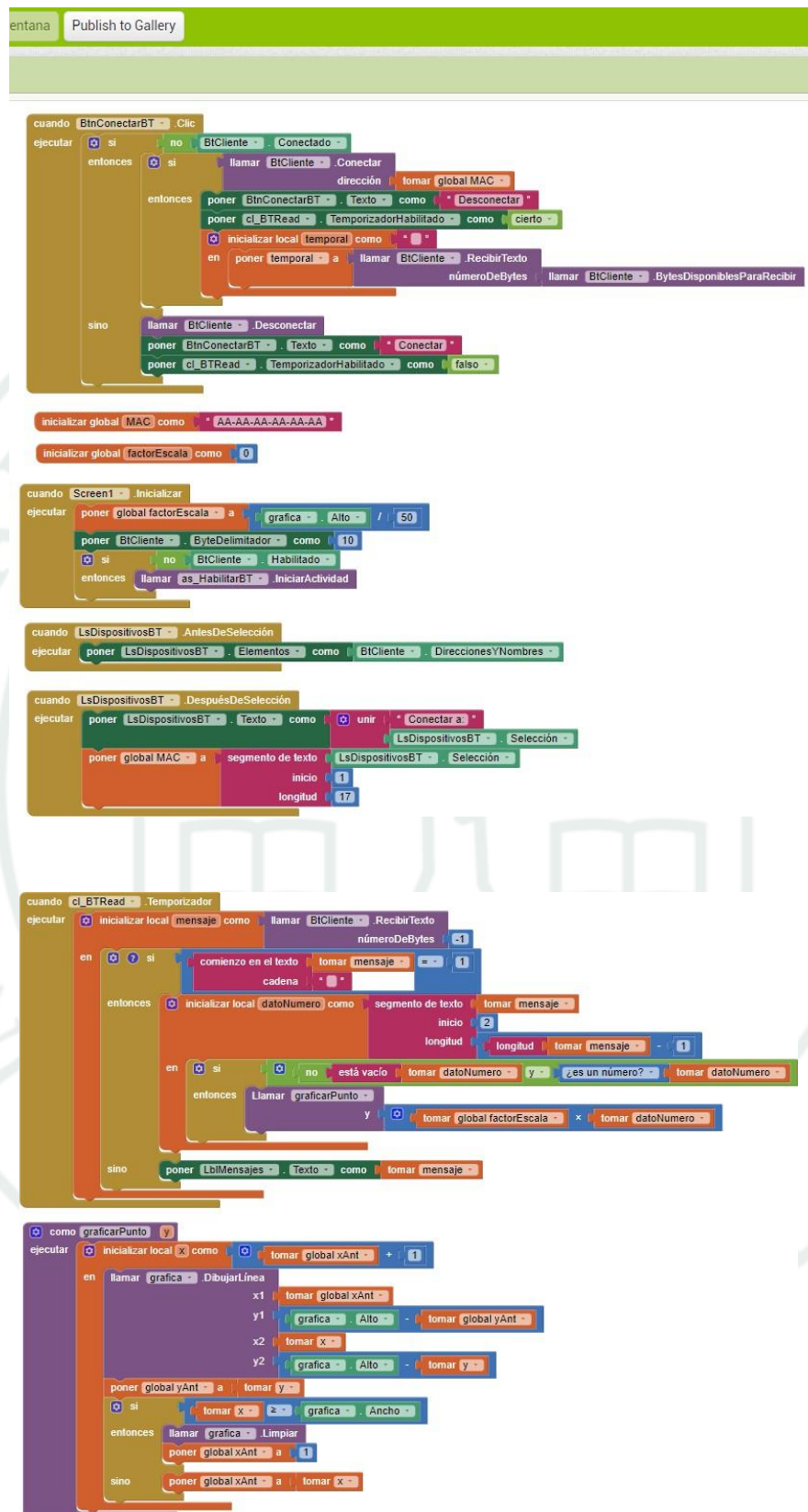
146      '----- Labeling values of the graph -----
147      Chart1.ChartAreas(0).AxisY.Maximum = 30
148      Chart1.ChartAreas(0).AxisY.Minimum = -5
149      Chart1.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled = True
150      '-----
151      '----- Labeling values of the graph -----
152      Chart2.ChartAreas(0).AxisY.Maximum = 1
153      Chart2.ChartAreas(0).AxisY.Minimum = -0.2
154      Chart2.ChartAreas("ChartArea1").AxisX.LabelStyle.Enabled = True
155      '-----
156  End Sub
157
158  Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
159
160      If Chart1.Series(0).Points.Count = ChartLimit Then
161          Chart1.Series(0).Points.RemoveAt(0)
162      End If
163      If Chart2.Series(0).Points.Count = ChartLimit Then
164          Chart2.Series(0).Points.RemoveAt(0)
165      End If
166
167  End Sub
168
169  Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click
170
171  End Sub
172 End Class
173
174
175

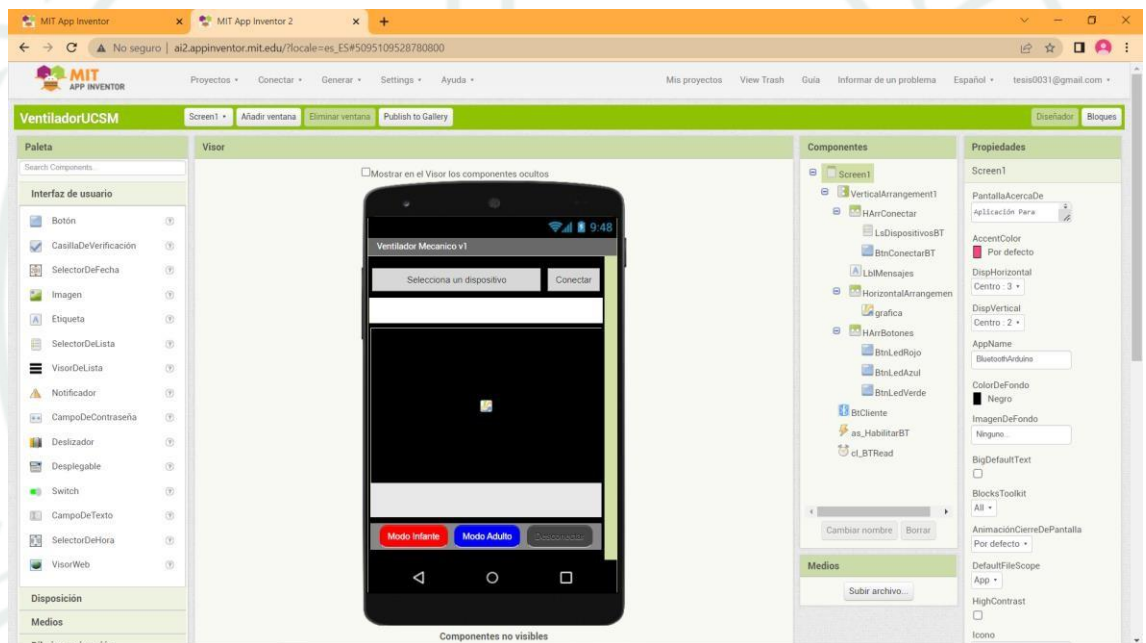
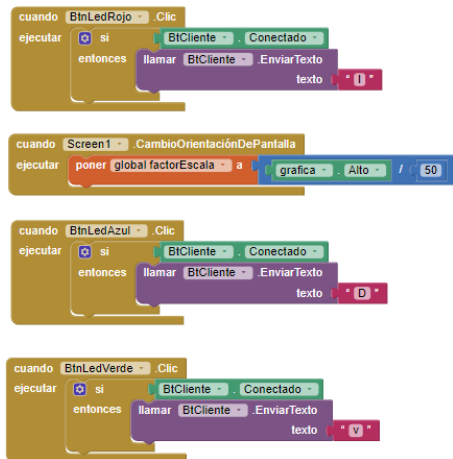
```

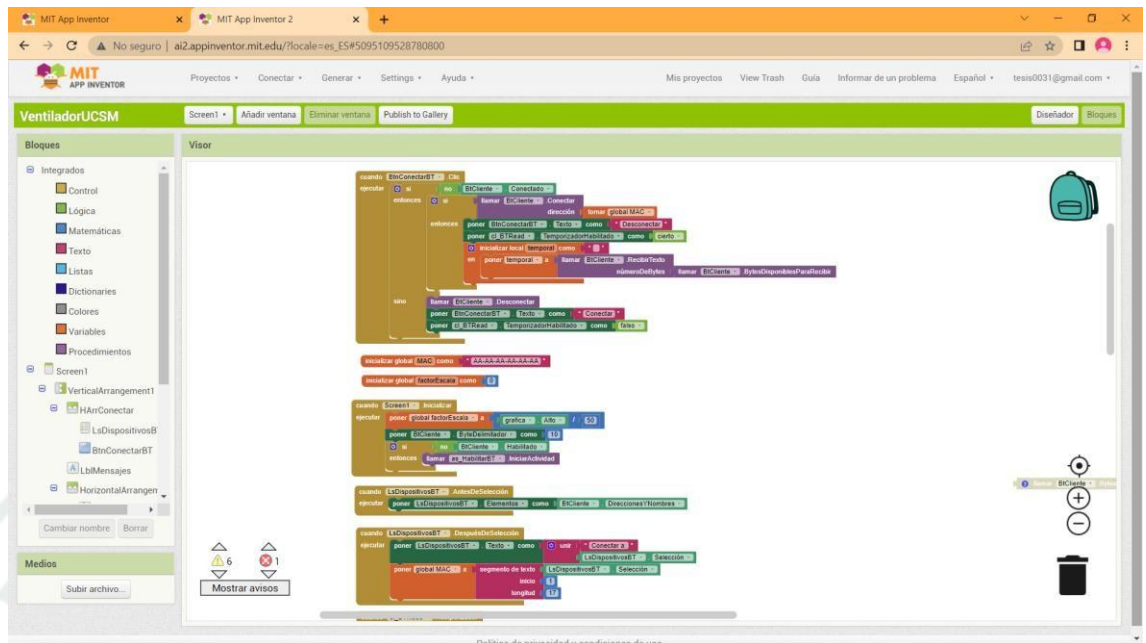
176
177
178
179
180
181
182
183
184
185



Anexo F4: Aplicación en APP Inventor







Anexo G: Prueba

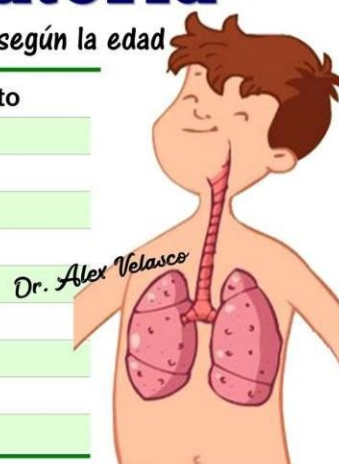
Anexo G1: Promedio Fisiológico de respiración

Según el Dr. Alex Velasco (@dr.alexvelasco) las respiraciones por minuto que presentan las personas se rigen a la siguiente imagen.

Frecuencia respiratoria

Valores Promedios de la Frecuencia Respiratoria normal según la edad

EDAD	Respiraciones por minuto
Neonato	30-50
0-5 meses	25-45
6-12 meses	20-40
1-3 años	20-35
3-5 años	20-30
6-12 años	15-30
13-18 años	14-20
Adultos	12-20
Vejez	12-16



Las cifras de la frecuencia respiratoria son muy variables se deben contemplar las variaciones fisiológicas como el ejercicio, el estrés, el ambiente, ascenso a grandes alturas, medicamentos que disminuyan la frecuencia respiratoria, la edad, la **fiebre** (se produce un aumento de **4 a 6** respiraciones por cada °C de ascenso de la temperatura), enfermedades y otras condiciones médicas.

Alteraciones de la Frecuencia Respiratoria

TAQUIPNEA	Aumento de la frecuencia respiratoria	> 20 respiraciones por minuto
EUPNEA	Respiración normal y fácil	12 - 20 respiraciones por minuto
BRADIPNEA	Disminución de la frecuencia respiratoria	< 12 respiraciones por minuto
APNEA	Cese del flujo de aire de al menos 20 segundos de duración	

Respiración rápida Es por lo general, el primer signo de dificultad respiratoria en lactantes y niños pequeños

Si el niño tiene	El niño tiene respiración rápida si usted cuenta
Menos de 2 meses	60 respiraciones o mas por minuto
2 a 11 meses	50 respiraciones o mas por minuto
12 meses a < 5 años	40 respiraciones o mas por minuto



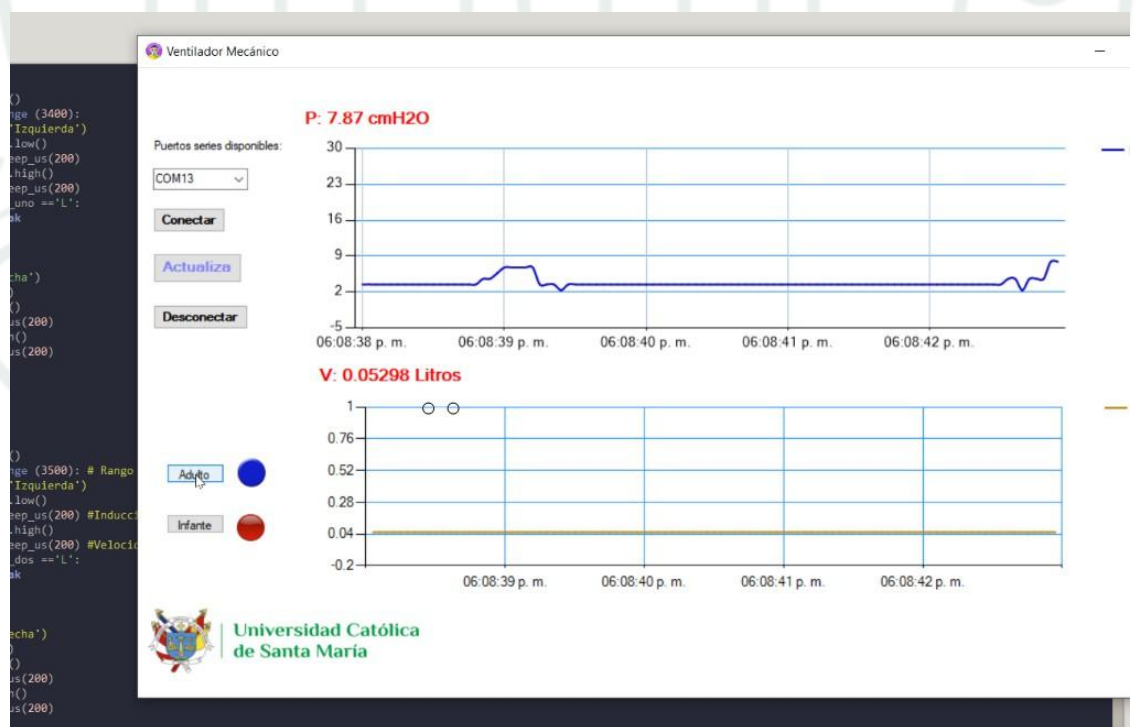
Anexo G2: Tabla experimental

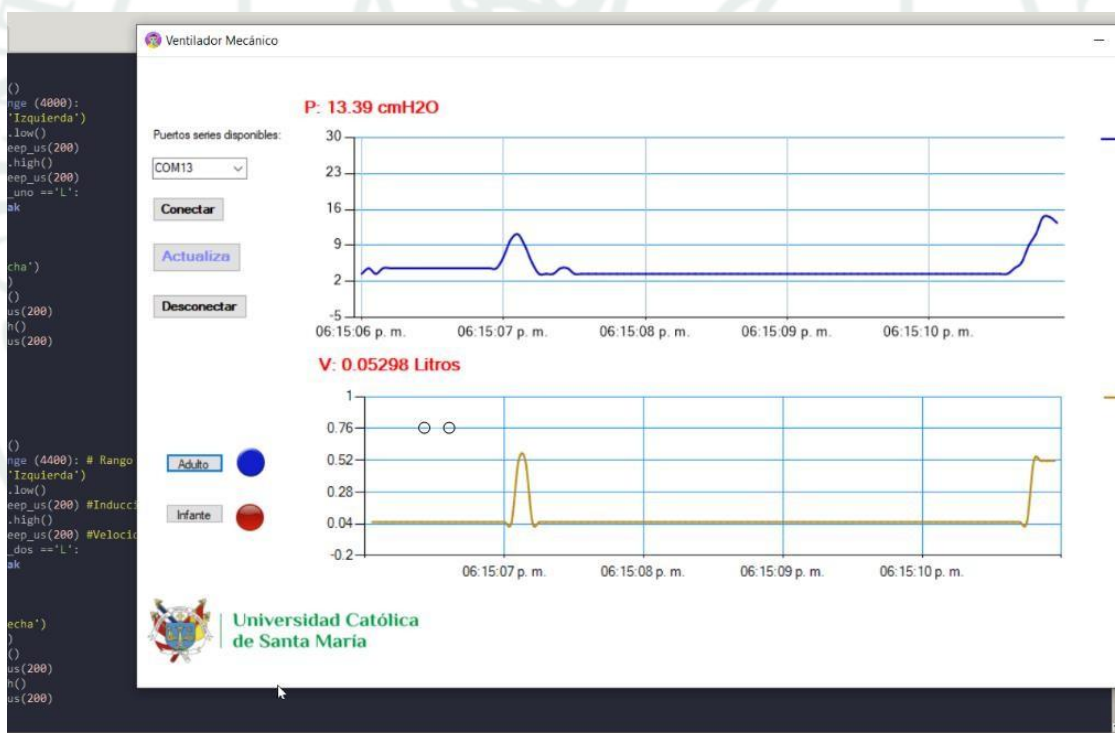
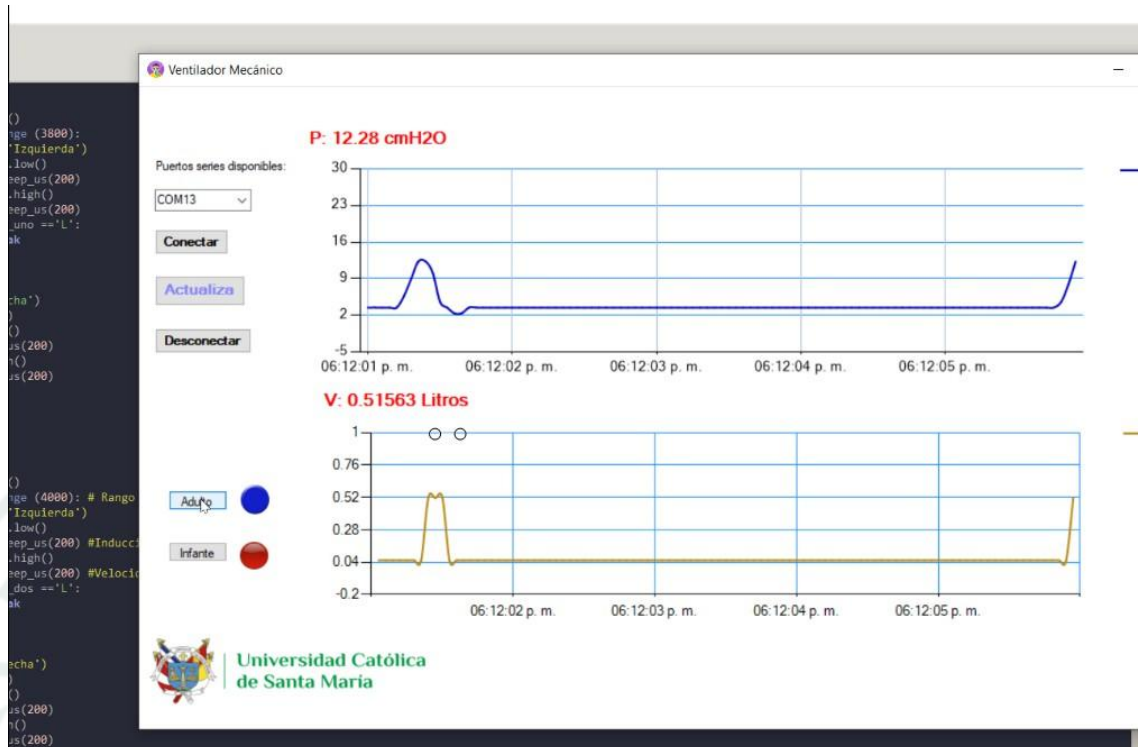
Configuración TB6600										
S1 S2 S3 S4 S5 S6	Paso	Rango	Ruido(dB)	Periodo(hora)		Segundos	Pre (cmH2O)	rpm	Promedio Seg	
ON OFF OFF ON OFF ON	500	1500	75	04:47:21	04:47:24	3.5	5.67	17	22.2	
		1700	74	04:48:22	04:48:27	4.2	6.77	14	4.44	2.22
		1600	74	04:51:35	04:51:39	4.5	7.87	13	Promedio Pre	
		1800	70	04:54	05:54:11	5	8.97	12	37.15	7.43
		2000	70	04:55:59	04:56:04	5	7.87	12		
									Promedio seg	
OFF ON ON ON OFF ON	400	1700	70	05:42:47	05:42:51	3.5	6.77	17		
		1800	70	05:43:30	05:43:34	4	7.87	15		
		1900	64	05:46:12	05:46:16	3.8	7.87	16		1.91
		2100	64	05:45:59	05:46:04	4	8.97	15		Promedio pre
		2200	59	05:48:51	05:48:55	3.8	7.87	16	39.35	7.87

ON OFF ON ON OFF ON	300	2200	64	05:53:26	05:53:30	3.4	5.67	18	Promedio seg	
		2400	67	05:53:36	05:53:39	2.6	6.77	23	19.2	
		2500	64	05:57:28	05:57:32	4.5	7.87	13	3.84	1.92
		2700	64	05:56:43	05:56:47	4.2	7.87	14	Promedio pre	
		3000	57	06:02:46	06:02:51	4.5	8.97	13	37.15	7.43
ON ON OFF ON OFF ON	200	3400	63	06:07:59	06:08:03	3.6	6.77	17	Promedio seg	
		3500	58	06:08:39	06:08:43	3.8	7.87	16	19.4	
		3800	57	06:12:01	06:12:06	4.5	12.28	13	3.88	1.94
		4000	57	06:12:07	06:12:11	4	13.39	15	Promedio pre	
		4400	54	06:16:00	06:00:04	3.5	14.49	17	54.8	10.96

Anexo G3: Captura de pantalla de la tabla experimental

Capturas de pantalla de la interfaz gráfica de usuario GUI





Captura de pantalla de ruido generado por el ventilador mecánico.





Anexo H: Formulario de evaluación y encuesta

Autoevaluación de Requisitos e especificaciones mínimas para la fabricación de un ventilador mecánico para pacientes con dificultad respiratoria, de la ciudad de Arequipa.									
Nombre:					 Universidad Católica de Santa María				
Edad:									
Nacionalidad:									
Cargo:		Medico:		Enfermera:		Paciente:			
Marcar la importancia de cada pregunta: 1 - No es importante en absoluto 2 - Poco importante 3 - Neutral 4 - Importante 5 - Muy importante									
a) ¿qué tan importante consideras el diseño compacto en un ventilador?	1	2	3	4	5				
b) ¿Qué tan importante consideras la transportabilidad de un ventilador mecánico en términos de su facilidad de movimiento, peso y tamaño?	1	2	3	4	5				
c) ¿qué tan importante consideras la calibración del prototipo en términos de su precisión y exactitud?	1	2	3	4	5				
d) ¿Qué tan importante consideras el tiempo de vida de un ventilador mecánico?	1	2	3	4	5				
e) ¿Qué tan importante consideras el mantenimiento que se debe realizar en un ventilador mecánico en un periodo de un año y tres meses?	1	2	3	4	5				
f) ¿Qué tan importante consideras la calidad del funcionamiento de un ventilador mecánico?	1	2	3	4	5				
g) ¿Qué tan importante consideras el monitoreo y la visualización de datos en un ventilador mecánico?	1	2	3	4	5				
h) ¿Qué tan importante consideras la calidad del funcionamiento de un ventilador mecánico?	1	2	3	4	5				
i) ¿Qué tan importante consideras el monitoreo y la visualización de datos en un ventilador mecánico?	1	2	3	4	5				
j) ¿Qué señales de emergencia consideras importantes que tenga un ventilador mecánico?	1	2	3	4	5				
k) ¿Qué tan importante consideras la seguridad del ventilador mecánico para los pacientes?	1	2	3	4	5				
l) ¿Qué tan importante consideras que un ventilador mecánico no genere ruido excesivo?	1	2	3	4	5				
m) ¿Qué tan importante consideras el costo operativo de un ventilador mecánico en términos de su consumo de corriente?	1	2	3	4	5				
n) ¿Qué tan importante es el costo de mantenimiento de un ventilador mecánico sea fácil y sencillo de realizar por cualquier persona?	1	2	3	4	5				

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE VENTILADOR MECÁNICO OPERABLE MANUAL Y REMOTAMENTE PARA LA ATENCIÓN DE PACIENTES CON PROBLEMAS RESPIRATORIOS



Un ventilador mecánico es un dispositivo que ayuda a las personas con problemas respiratorios a respirar. El ventilador proporciona aire a los pulmones de la persona, lo que ayuda a mantener los niveles de oxígeno en la sangre. El ventilador también puede ayudar a eliminar el dióxido de carbono de los pulmones.

*Obligatorio

Dirección de correo electrónico *

- ¿Ha escuchado hablar de la ventilación mecánica y su impacto en el momento de salvar vidas? *

Marca solo un rectángulo.

SI ☐

NO ☐

- ¿Cree usted que sea de vital importancia la construcción de ventiladores mecánicos para combatir la emergencia sanitaria ocasionado por el COVID-19? *

- SI ☐

- NO ☐

Según su punto de vista, elija una de las 5 alternativas indicando la importancia relativa de los siguientes aspectos en el desarrollo de un respirador automático, asignando el número 1 al de menor importancia y el número 5 al aspecto más crucial.

1. ¿Cuál es su opinión sobre la relevancia de la facilidad de uso en el respirador?

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

2. ¿Cuál es su evaluación sobre la importancia de garantizar la seguridad del paciente al utilizar el respirador?

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

3. ¿Qué tan importante cree usted que sea la facilidad de esterilización del equipo? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

4. ¿Qué grado de importancia atribuye usted al control de la presión y la duración del ciclo de respiración administrado al paciente por el equipo?

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

5. ¿Qué tan importante es para usted el tamaño del ventilador mecánico? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

6. ¿Considera usted que es importante que el equipo sea de bajo costo? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

7. Acerca de la fácil transportación, ¿qué importancia tiene para usted? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

8. Es importante para usted que el equipo requiera de un fácil mantenimiento? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

9. La apariencia o estética del equipo, que tanto es influyente según su criterio?

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

10. Qué tan importante es la disponibilidad de repuestos para la fabricación del equipo? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

11. ¿Cuán importante cree usted que es el fácil desmontaje del equipo en caso de alguna falla?

Marca solo un rectángulo.

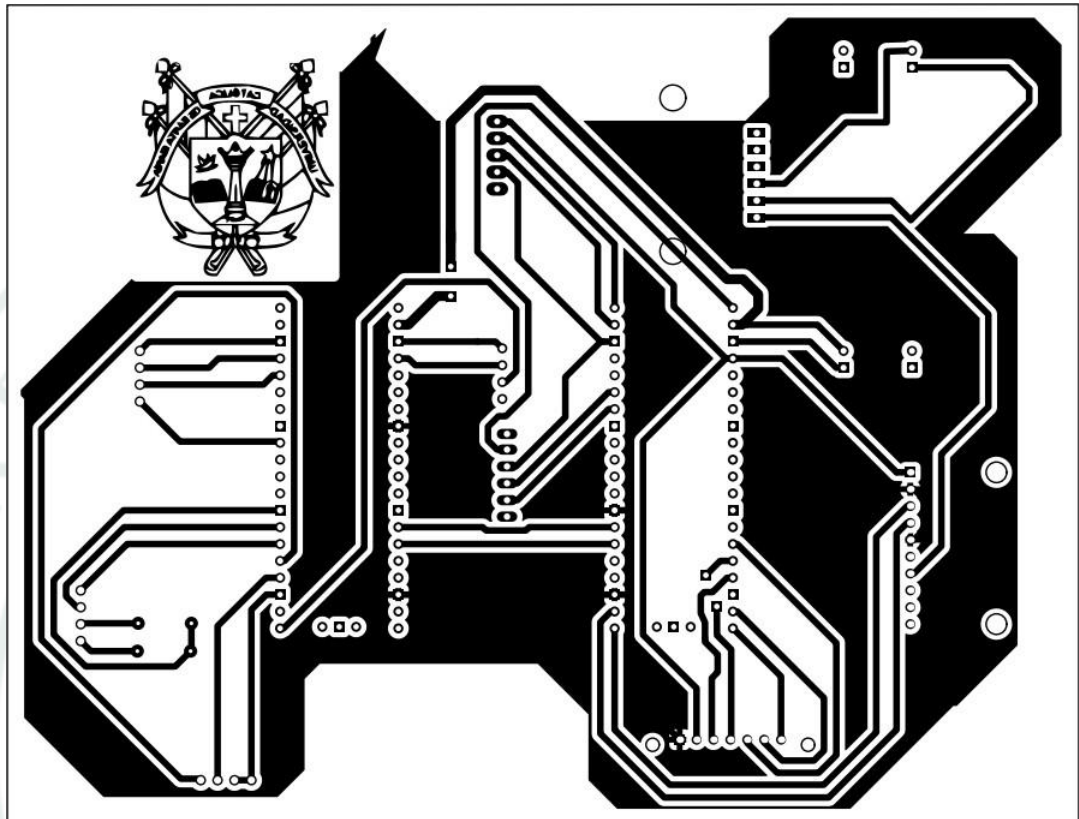
menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

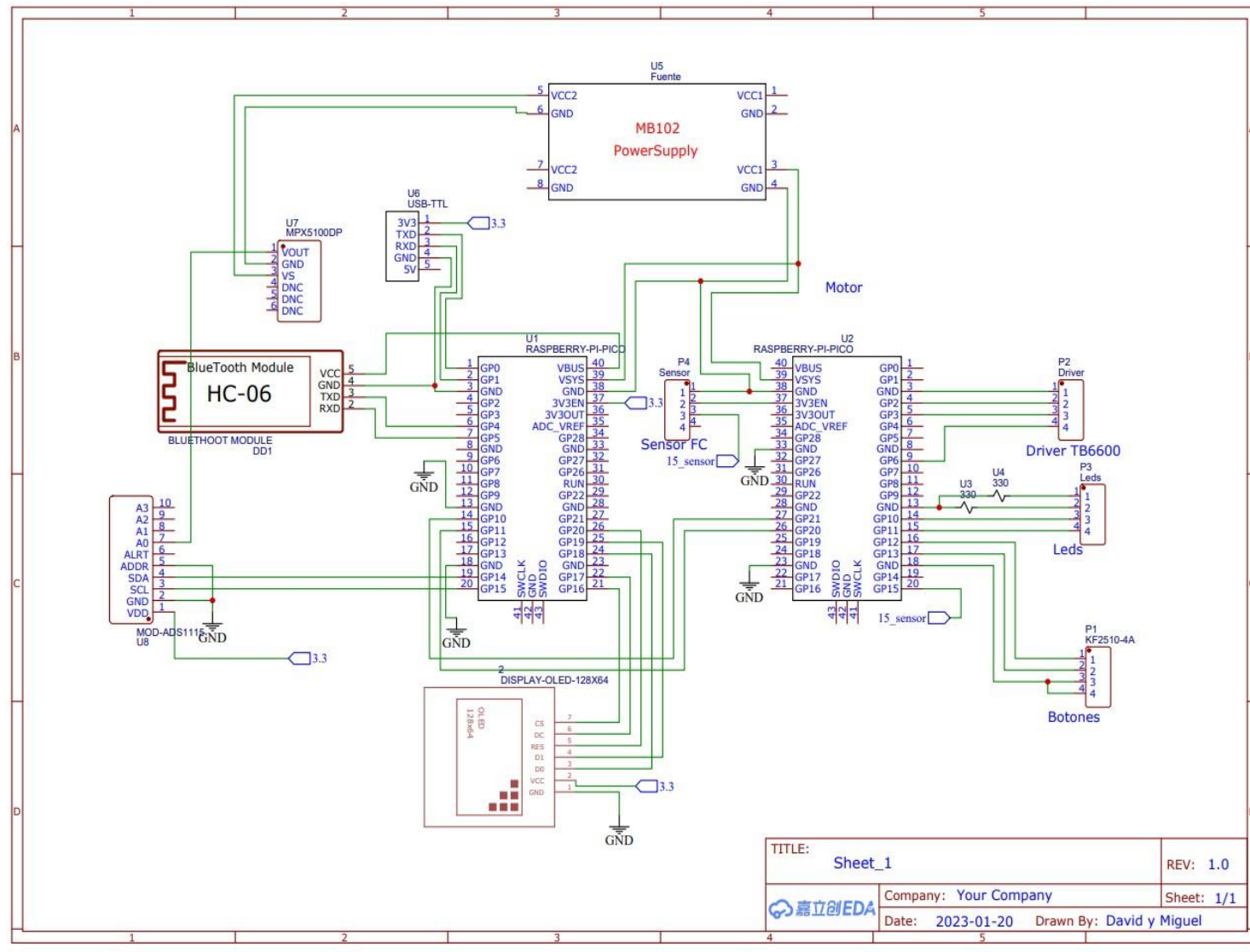
12. Es relevante que el equipo tenga un sistema de alarma, que alerte al médico en caso de alguna emergencia? *

Marca solo un rectángulo.

menor importancia	1	2	3	4	5	mayor importancia
-------------------	---	---	---	---	---	-------------------

Anexo I: Circuito PCB y plano esquemático





This is a preview - click here to buy the full publication



IEC 60601-1

Edition 3.1 2012-08
CONSOLIDATED VERSION

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Medical electrical equipment –
Part 1: General requirements for basic safety and essential performance**

**Appareils électromédicaux –
Partie 1: Exigences générales pour la sécurité de base et les performances
essentielles**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 11.040

ISBN 978-2-8322-0331-8

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

© Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

A continuación, se describen las especificaciones técnicas que deben contar los ventiladores mecánicos o ventiladores portátiles, indicadas por la Organización Mundial de la Salud en el documento *"Lista de Dispositivos Médicos Prioritarios en el contexto del COVID-19, (recomendaciones provisionales, 06 abril 2020) 3ra Versión"*¹:

- Volumen corriente 5 a 2000 ml.
- Presión (inspiratoria) hasta 80 cm H₂O.
- Flujo (inspiratorio) 2 a 180 L/min.
- Frecuencia respiratoria: hasta 150 respiraciones por minuto.
- CPAP/PEEP hasta 20 cm H₂O.
- Presión de soporte de hasta 45 cm H₂O.
- FIO₂ entre el 21 y el 100%.
- Tiempos inspiratorios y espiratorios de al menos 0,1 s y 10 s respectivamente.
- Relación I:E al menos de 1:1 a 1:4.
- Modos de ventilación: a) Control de volumen. B) Control de presión. C) Soporte de presión. D) Modo de asistencia/control f) CPAP/PEEP.
- Filtro bacteriano, viral, electrostático, EPA de alta eficacia.
- Alarmas requeridas: FiO₂, volumenmínusculo, presión, PEEP, apnea, oclusión, alta tasa de respiración, desconexión.
- Se requieren alarmas del sistema: corte de energía, desconexión de gas, batería baja, ventilación inoperativa, autodiagnóstico.
- Si se incorpora la función de silenciamiento de alarma, debe ser temporal y claramente mostrada cuando se activa.
- Las relaciones de mezcla de oxígeno suministradas externamente por aire.
- Rango de presión de suministro de gas de entrada (O₂) al menos 35 a 65 PSI.
- Utilizar mecanismos de humidificación pasiva siempre que sea posible para reducir el riesgo para el personal de salud.
- En contextos donde no esté garantizada la disponibilidad de sistema adecuado de aire comprimido (hospitales de campaña, etc.) puede optarse por compresor de aire médico integrado a unidad, con filtro de entrada.
- Incluir batería que permita autonomía de hasta 1 hora. El tiempo de recarga total no debe superar las 6 horas.

Anexo K: Google Colab

Se adjunta el código empleado para la búsqueda de la mejor opción de operación para el modo infante y adulto.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
import pydotplus
from IPython.display import Image

# Definir los datos
datos = [
    (500, 1500, 75, 3.5, 5.67, 17),
    (500, 1600, 74, 4.2, 6.77, 14),
    (500, 1700, 74, 4.5, 7.87, 13),
    (500, 1800, 70, 5, 8.79, 12),
    (500, 2000, 70, 5, 7.87, 12),
    (400, 1700, 70, 3.5, 6.77, 17),
    (400, 1800, 70, 4, 7.87, 15),
    (400, 1900, 64, 3.8, 7.87, 16),
    (400, 2100, 64, 4, 8.97, 15),
    (400, 2200, 59, 3.8, 7.87, 16),
    (300, 2200, 64, 2.5, 5.67, 24),
    (300, 2400, 67, 2.6, 6.77, 23),
    (300, 2500, 64, 3.6, 7.87, 17),
    (300, 2700, 64, 4.2, 7.87, 14),
    (300, 3000, 57, 4.5, 8.97, 13),
    (200, 3400, 63, 3.6, 6.77, 17),
    (200, 3500, 58, 3.8, 7.87, 16),
    (200, 3800, 57, 4.2, 12.28, 14),
    (200, 4000, 57, 4, 13.39, 15),
    (200, 4400, 54, 3.5, 14.49, 17)
]

# Crear un DataFrame de pandas con los datos
df = pd.DataFrame(datos, columns=['Paso', 'Rango', 'dB', 'Tiempo',
    'Presión', 'rpm'])

# Definir las características (X) y la variable objetivo (y)
X = df[['rpm']]
y = (df['rpm'] >= 21.6) & (df['rpm'] <= 23.4) # Clasificar como
"Cercano a 22.5" si está en el rango  $\pm 0.9$  alrededor de 22.5 o 14

# Entrenar el modelo de árbol de clasificación
modelo = DecisionTreeClassifier(max_depth=3)
modelo.fit(X, y)

# Generar el diagrama del árbol
dot_data = export_graphviz(modelo, out_file=None,
    feature_names=['RPM'], class_names=['No cercano a 22.5', 'Cercano a
    22.5'], rounded=True, filled=True)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
import pydotplus
from IPython.display import Image

# Definir los datos
datos = [
    (500, 1500, 75, 3.5, 5.67, 17),
    (500, 1600, 74, 4.2, 6.77, 14),
    (500, 1700, 74, 4.5, 7.87, 13),
    (500, 1800, 70, 5, 8.79, 12),
    (500, 2000, 70, 5, 7.87, 12),
    (400, 1700, 70, 3.5, 6.77, 17),
    (400, 1800, 70, 4, 7.87, 15),
    (400, 1900, 64, 3.8, 7.87, 16),
    (400, 2100, 64, 4, 8.97, 15),
    (400, 2200, 59, 3.8, 7.87, 16),
    (300, 2200, 64, 2.5, 5.67, 24),
    (300, 2400, 67, 2.6, 6.77, 23),
    (300, 2500, 64, 3.6, 7.87, 17),
    (300, 2700, 64, 4.2, 7.87, 14),
    (300, 3000, 57, 4.5, 8.97, 13),
    (200, 3400, 63, 3.6, 6.77, 17),
    (200, 3500, 58, 3.8, 7.87, 16),
    (200, 3800, 57, 4.2, 12.28, 14),
    (200, 4000, 57, 4, 13.39, 15),
    (200, 4400, 54, 3.5, 14.49, 17)
]

# Crear un DataFrame de pandas con los datos
df = pd.DataFrame(datos, columns=['Paso', 'Rango', 'dB', 'Tiempo',
    'Presión', 'rpm'])

# Definir las características (X) y la variable objetivo (y)
X = df[['rpm']]
y = (df['rpm'] >= 21.6) & (df['rpm'] <= 23.4) # Clasificar como
"Cercano a 22.5" si está en el rango  $\pm 0.9$  alrededor de 22.5

# Entrenar el modelo de árbol de clasificación
modelo = DecisionTreeClassifier(max_depth=3)
modelo.fit(X, y)

# Generar el diagrama del árbol
dot_data = export_graphviz(modelo, out_file=None,
    feature_names=['RPM'], class_names=['No cercano a 22.5', 'Cercano a
    22.5'], rounded=True, filled=True)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())

```

Anexo L: Constancia de evaluación de fiabilidad del prototipo



**CENA
SAC**

"AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA ECONOMÍA PERUANA"

Constancia de evaluación de fiabilidad del prototipo de ventilador mecánico por profesionales del sector salud

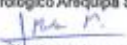
Por medio de la presente, se deja constancia de que el prototipo de ventilador mecánico, desarrollado por los estudiantes Álvarez Gómez David Joel y Teran Vargas Miguel Ángel, de la carrera profesional de Ingeniería Electrónica de la Universidad Católica de Santa María, ha sido evaluado en términos de fiabilidad, desempeño y estabilidad operativa, contando con la validación de profesionales especializados en el área de la salud.

Dicha evaluación se realizó el día 15 de setiembre del 2025 en el Centro Nefrológico Arequipa S.A.C. – CENA S.A.C. sede Las Lilas, obteniéndose observaciones y resultados favorables que respaldan el funcionamiento del prototipo en relación con los parámetros establecidos para dispositivos de asistencia respiratoria.

Las recomendaciones técnicas y la relación de profesionales participantes se detallan en la sección complementaria, la cual forma parte integrante de la presente constancia.

Se expide la presente a solicitud de los interesados, para los fines académicos y administrativos que estimen pertinentes, no constituyendo autorización para uso clínico ni reemplazo de certificaciones correspondientes.

Arequipa, 16 de setiembre del 2025

CENA S.A.C.
Centro Nefrológico Arequipa S.A.C.

Lizeth Morales Cusiquispe
ADMINISTRADORA

Psje Las Lilas 105 Yanahuara – Arequipa – Teléfonos: +51(1) 6403314 +51(1) 6408012
E-mail: cenape@cidlatam.com

SECCIÓN COMPLEMENTARIA
Recomendaciones y Profesionales Participantes

Recomendaciones

1. Implementar una batería que permita el funcionamiento sin necesidad de conexión directa a la red eléctrica.
2. Incorporar una pantalla de mayor tamaño en el control local, para facilitar la visualización de gráficas en función del tiempo.
3. Reducir el nivel de ruido (decibelios) generado por el prototipo.
4. Implementar un sistema de ventilación que evite el sobrecalentamiento del equipo.
5. Incorporar una alarma de advertencia ante excesos de presión.
6. Elaborar un manual de uso y realizar la capacitación correspondiente para su correcta operación.

Profesionales Participantes en la Evaluación

1. Jackeline Juliana Rodríguez Paco – Médico Nefrólogo
2. Sidney Arteta Borda – Médico Nefrólogo
3. Patricio Miguel Charres Coaguila – Médico Nefrólogo
4. Eliana Soledad Pauca Bautista – Licenciada en Enfermería
5. Yesenia Bedregal Callata – Licenciada en Enfermería
6. Gloria Abdulia Cuba de Hospinal – Licenciada en Enfermería
7. Andrea Isabel Hilasaca Flores – Técnica en Enfermería
8. Edith Nerle Mollo Llamoca – Técnica en Enfermería
9. Eduar Macdonald Carpio Maquerhua – Técnico en Enfermería
10. Giovanna Estela Zumaran Salazar – Técnica en Enfermería
11. Cristhian Hilasaca Gómez – Técnico en Mantenimiento de Máquinas
12. Alexander Gutiérrez Chávez – Técnico en Mantenimiento de Máquinas
13. Ángel Machaca Mamani – Técnico en Mantenimiento de Máquinas

Anexo M: Especificaciones técnicas y físicas del prototipo

Sistema de alimentación

El prototipo presenta una arquitectura de alimentación multinivel, basada en dos fuentes principales de energía:

- **Fuente primaria:** 24 V DC (alimentación del sistema de potencia)
- **Fuente secundaria:** 12 V DC (subsistemas intermedios)

A partir de estas fuentes, se generan niveles de tensión inferiores mediante reguladores de voltaje.

Niveles de tensión del sistema

El sistema opera con los siguientes niveles de tensión:

- **24 VDC:** alimentación del driver TB6600.
- **12 VDC:** alimentación auxiliar del sistema.
- **5 VDC:** alimentación de periféricos (pantalla OLED, ADS1115, LEDs).
- **3.3 VDC:** alimentación de microcontroladores y módulo Bluetooth.

La conversión de voltaje se realiza mediante reguladores tipo step-down integrados en el sistema

Componentes electrónicos principales

- 2 microcontroladores Raspberry Pi Pico
- Módulo Bluetooth (comunicación inalámbrica)
- Módulo USB (comunicación alámbrica)
- Convertidor analógico-digital ADS1115
- Sensor de presión MPX5010DP
- Pantalla OLED 128×64
- Driver de potencia TB6600
- Motor paso a paso NEMA 17
- Elementos de interfaz como LEDs

Consumo de corriente estimado

Componente	Corriente aproximada
2 × Raspberry Pi Pico	100 mA
Módulo Bluetooth	30 mA
ADS1115	1 mA
Sensor MPX5010DP	10 mA
Pantalla OLED	80 mA
LEDs (2)	30 mA
Driver + motor NEMA 17	1.5 – 2.5 A
Corriente total estimada	2 – 3 A

Potencia del sistema

Tomando como referencia la fuente principal:

$$P = V \times I = 24 \text{ V} \times 2.5 \text{ A} = 60 \text{ W}$$

Potencia estimada del sistema 50 – 70 W

Sistema de control

El control del prototipo se realiza mediante dos microcontroladores, los cuales se encargan de:

- Adquisición de señales del sensor de presión mediante el ADS1115.
- Procesamiento de datos.
- Generación de señales de control para el driver TB6600.
- Gestión de la interfaz de usuario como pantalla oled y LEDs.
- Comunicación con dispositivos externos mediante Bluetooth y USB.

Interfaces de comunicación

- Comunicación I2C (ADS1115).
- Comunicación SPI (Pantalla OLED).
- Comunicación UART (comunicación inalámbrica y alámbrica).
- Comunicación digital (final de carrera, LEDs).

Especificaciones de la bolsa AMBU

La bolsa tipo AMBU (Artificial Manual Breathing Unit) es un dispositivo médico utilizado para proporcionar ventilación manual asistida mediante la compresión de una cámara de aire. En el presente prototipo, este elemento es adaptado para su accionamiento mecánico mediante la estructura impulsora de aire, permitiendo la administración controlada de aire al paciente.

Especificaciones funcionales

- Volumen **de la bolsa:** 1000 ml – 1600 ml (adulto típico).
- Volumen **Tidal suministrado:** 300 ml – 700 ml (dependiente de la compresión).
- Frecuencia **de operación:** 10 – 30 respiraciones por minuto.
- Presión **de operación:** hasta aproximadamente 45 cmH₂O.

Componentes principales

- Bolsa autoinflable (cámara de aire).
- Válvula unidireccional.
- Conector para mascarilla.
- Válvula de alivio de presión.
- Entrada para suministro de oxígeno.

Especificaciones físicas de la estructura

El diseño estructural del prototipo ha sido desarrollado considerando criterios de portabilidad y resistencia mecánica.

- Dimensiones: 80 cm × 60 cm × 15 cm
- Peso aproximado: 8 – 10 kg
- Materiales empleados:
 - Estructura: Filamento ABS con espesor de 1.75 mm
 - Componentes de soporte: Madera

Tabla resumen de especificaciones

Parámetro	Especificación
Voltaje de operación	24 V DC
Corriente nominal	2.5 A
Frecuencia respiratoria	14 – 23 rpm
Volumen Tidal	300 – 700 ml
Material estructural	Filamento ABS de 1.75 mm
Peso	8 – 10 kg

Las especificaciones técnicas presentadas corresponden a un prototipo experimental, por lo que su implementación está sujeta a procesos de validación, calibración y cumplimiento de normativas biomédicas antes de su aplicación en entornos clínicos reales. El prototipo presenta un sistema electrónico de arquitectura multinivel, en el cual se integran etapas de control, adquisición de datos y potencia.

Parámetros de rendimiento

Parámetro	Adulto	Niño	Unidad
Frecuencia respiratoria	14 – 14.71	21.66 – 22.5	respiraciones/min
Volumen tidal	420	310	ml
Presión pico	19.96 – 20	15 – 16	cmH ₂ O
PEEP	5 – 25	5 – 25	cmH ₂ O
Ruido	57	67	dB
Tiempo de funcionamiento continuo		Hasta 2	
Tiempo de respuesta del sistema		≤ 1 segundo	

Anexo N: Condiciones de operación y parámetros

Condiciones de operación

El funcionamiento adecuado del prototipo está garantizado bajo las siguientes condiciones ambientales:

- Temperatura de operación: 15 °C a 35 °C.
- Humedad relativa: 30 % a 80 % (debido a los ambientes donde se usa).
- Uso en entornos clínicos o de emergencia.
- Compatible con aire ambiente u oxígeno suplementario.

Parámetros de rendimiento

El desempeño del sistema se define en función de los siguientes parámetros operativos:

- Frecuencia respiratoria: 14 a 20 respiraciones por minuto.
- Volumen Tidal estimado: 300 a 700 ml.
- Tiempo de respuesta del sistema: menor o igual a 1 segundo.
- Tiempo de funcionamiento continuo: hasta 2 horas.

Sistemas de seguridad

El prototipo incluye mecanismos básicos de seguridad para garantizar su operación controlada:

- Botón de parada de emergencia.
- Indicadores visuales del estado del sistema.
- Válvula unidireccional.
- Válvula de alivio de presión.